Mapping with a ground robot in GPS denied and degraded environments

John G. Rogers III, Jonathan R. Fink, and Ethan A. Stump

Abstract—A robot system operating in an unknown environment must be able to track its position to perform its mission. Vehicles with a consistent view of the sky, e.g., aerial or water surface platforms, can reliably make use of GPS signals to correct accumulated error from inertial measurements and feature-based mapping techniques. However, ground robots that must operate across a wide range of environments suffer from additional constraints which degrade the performance of GPS such as multipath and occlusion. In this paper, we present a methodology for incorporating GPS measurements into a feature-based mapping system for two purposes: providing georeferenced coordinates for high-level mission execution and correcting accumulated map error over long-term operation. We will present both the underlying system and experimental results from a variety of relevant environments such as military training facilities and large-scale mixed indoor and outdoor environments.

I. INTRODUCTION

One of the most basic capabilities for autonomous robotic vehicles is the ability to understand and execute high-level tasks relating to navigation or positional control in their operational environment. On top of this capability, a wide range of missions can be specified, e.g., patrol, surveillance, exploration, intelligence gathering, etc. In certain applications, for instance, warehouse operation or manufacturing, the operating environment is known a priori and straightforward techniques for estimating location have been extremely successful [23]. However, as robots are tasked to operate in a wider range of personal, commercial, and military environments, it is impractical or impossible to supply every system with an accurate map.

Thus, autonomous robotic vehicles must be able to build a world model or map of their environment in order to properly parse and then execute navigation tasks. In some outdoor environments, Global Positioning System (GPS) measurements provide a reference to a global coordinate system which is useful for mission specification and execution. GPS measurements can also provide important global constraints which can be used to correct accumulated error. The current OmniMapper system has been used to integrate information from a variety of sensor sources such as Velodyne 32E3D laser scanners, Microsoft Kinect 3D cameras, Hokuyo 2D laser scanners which are mounted on a tilting platform, as well as planar, unactuated 2D laser scanners.

The OmniMapper system has been tested in a variety of settings such as in office buildings, at military training facilities and in large scale in-/out-door environments. This paper will present the conditions when GPS can be used with the OmniMapper to improve mapping performance, together with experimental results from a variety of settings.

Some of the relevant related work in robot mapping will be described in Section II. The experimental methodology and algorithms will be described in Section III. The experimental settings and results will be presented in Section IV. Finally, conclusions will be presented in Section V.

II. RELATED WORK

GPS can be used by a mobile ground robot for localization. Due to some of the limitations of GPS on ground robots which were listed in the previous section, this signal is typically coupled with another sensory modality such as visual odometry [1]. This type of sensory feedback is essentially intrinsic to the platform as it measures only relative motion.
Another approach to handle the GPS limitations listed above is to couple it with another type of extrinsic measurement. GPS measurements are augmented with localization in aerial imagery in [8] to provide accurate global robot localization. We will show that GPS measurements can be used in a modern graph-based mapping system to improve map accuracy and provide a global frame of reference.

The techniques described in this paper address the problem of simultaneous localization and mapping (SLAM). SLAM is one of the core research problems in mobile robotics [6]. Currently, mobile robots are able to build maps in a variety of environments in size up to that of an entire city [2]. The techniques used in this paper are based upon square-root smoothing and mapping (√SAM) [4], [5]. √SAM uses repeated nonlinear least-squares optimization to compute the robot’s trajectory and landmark locations from a set of relative pose and landmark measurements.

III. METHODOLOGY

The approach described in this paper uses a mapping system to incorporate and validate GPS information to provide the robot with an accurate global localization. The GPS information is also used to generate maps of mixed indoor and outdoor environments which can be shared with other robots or human operators.

A. Mapping system

The OmniMapper system which was introduced above makes use of an implementation of √SAM provided by the GTSAM library developed at Georgia Tech. OmniMapper has been used in prior work to build maps of indoor and outdoor environments [17], to determine mapping performance with sensory degradation [11], for multi-robot mapping [12], [9], and multi-robot mapping with heterogenous sensory modalities [10]. The system has also been used to support semantic mapping, where additional high-level information is included within a map as it is observed by the robot. The first type of high level information we have included is called virtual measurements [22], where parallel and perpendicular constraints are included between walls which exhibit these relationships, and visual features which appear to be coplanar with walls are constrained to lie upon them. This technique has also been used with object recognition for building semantic maps together with a powerful cue for loop closure in [16], and object recognition has been used with the map to perform place recognition [15], [14], [13].

In the approach described in this paper, our mapping system uses the GTSAM library to optimize a graph of relative robot pose measurements made with intrinsic odometry as well as 3D iterative closest point-based scan matching. This system also makes use of measurements between distant places during loop closure to correct accumulated map error. We have shown, in the work cited above, that this class of mapping system can perform adequately in many GPS-denied environments. In this paper, we will show how this system can be extended to incorporate GPS measurements, when available, to ground the map in a global reference frame and provide additional cues for the correction of long-term accumulated map error.

OmniMapper is a modularly constructed with a system of plugins corresponding to specific sensor modalities and feature types. This system was initially developed for use with feature-based mapping of lines, planes, and objects. More recently, we have developed feature-agnostic or featureless mapping plugins based upon Iterative Closest Point (ICP) [3] for more general use in cluttered or austere environments. The relative-pose and loop-closure measurement plugin is based upon Generalized-ICP [18] which uses point-to-plane correspondences and performs well on large sparse point clouds.

The ICP algorithm is a two-phase process which is repeated until convergence. This algorithm takes two point clouds as inputs, in our case these were generated with a 3D laser scanner. The first of these clouds is a reference cloud from the previous scan and the second is a new cloud from the current scan. In the first phase, putative point matches are determined by selecting for each point in the new cloud the closest point in the reference cloud. In the second phase, these putative point matches form a set of constraints which is optimized via a nonlinear estimation algorithm to determine the relative pose between these two clouds. This transformation is then applied to the input cloud. When the iteration is restarted with the putative correspondence selection, the point clouds have moved closer to convergence, therefore different correspondences might be selected. The iteration is repeated until the updated transform is small and the procedure has converged.

B. Measurements for mapping

The ICP plugin in OmniMapper contains two threads of computation. The first thread of computation processes new point clouds as they are produced from sensor data. These point clouds can be produced from many types of sensors. The sensor modalities which have been used with this plugin include Hokuyo UTM30 laser scanners which are mechanically actuated via Directed Perception pan-tilt units, Kinect type 3D cameras, and Velodyne 32E 3D LIDAR scanners. The second thread of computation looks for loop closures with the most recent point cloud and other point clouds which were collected along the robot’s past trajectory. These two threads provide pose measurement constraints which are used to compute the robot’s trajectory. This trajectory information is used to render all point cloud data into one large point cloud representing the entire map.

ICP-type algorithms are costly in terms of computation time. To maintain online operation, the resolution and density of the point clouds is reduced. The robot is able to approximately track its displacement using inertial and odometric measurements; these measurements provide a starting point for the ICP routine which is close to the correct solution, thereby reducing the number of iterations required. This inertial and odometric initial condition is provided in free spatial motion in SE(3); ICP then refines this initial condition to a final relative pose estimate in SE(3).

1881
example assembled point cloud is built from a robot in an
office building using OmniMapper with Generalized-ICP.

(a) A 3D view of a point cloud built up by a robot using the Generalized-ICP plugin with OmniMapper

(b) Example point cloud map built illustrating the result of loop closure on an outdoor run. Loop closure constraints are drawn as green lines between trajectory elements which are drawn as red arrows. Relative pose constraints solved by ICP are represented with blue lines between adjacent trajectory elements.

Fig. 1. OmniMapper maps built by finding relative poses in adjacent point-clouds and loop closures between distant trajectory elements.

A loop-closure thread looks for possible loop closures when the robot crosses its previous trajectory by trying to perform ICP between the point clouds observed at these locations. When the robot gets close to its previous trajectory, it chooses the closest location to attempt alignment between its current point cloud and that previous one. If these point clouds can be aligned, then a relative pose measurement factor is added to the graph between these two poses and the map is re-optimized. An example of loop closure by ICP-alignment of point clouds from non-adjacent trajectory elements can be seen in Figure 1(b). Here, green lines indicate the constraints inserted via the loop closure ICP routine.

The loop closure routine is unlikely to work if the robot has traveled a long distance before returning to this previous location along the trajectory. This is due to the fact that ICP routines must be initialized within a relatively close region of convergence to increase likelihood of success. In addition, ICP tends to foreshorten motion along corridors in the absence of constraining features along the direction of motion; as it is maximizing overlap between point clouds. The system incorporates GPS data to improve trajectories so they are close enough to perform loop closure. It should be noted that our system is able to perform mapping using sensor measurements and odometry alone. The addition of GPS measurements as an additional data source is used to enhance the mapper’s accuracy and establish a global frame of reference.

C. Reconciliation of global and relative reference frames

Until GPS measurements are incorporated into the map, the mapping process occurs in a reference frame that is relative to the initial pose of the robot, i.e., all inertial and ICP-based measurements only provide information about the change in relative pose. This presents a fundamental challenge when incorporating a GPS measurement: It is an under defined, i.e., 3D position without orientation, measurement that is relative to a global reference frame that is unknown with respect to the existing map reference frame. In theory, a pose-graph-based mapping method should be able to resolve the transformation from global to map reference frames. However, in practice we find that this optimization is sensitive to initial conditions and often converges to local minima.

Instead, we adopt a two-phase approach where we associate each GPS measurement with an optimized robot pose in the existing map frame. Given a minimum of two associated GPS measurements and map poses, we can perform an SVD-based estimation of the transform between the global GPS frame and the relative map frame. We note that since the GPS measurements are subject to considerable noise, a number of measurements are necessary to accurately estimate this transformation. Indeed, we continue to collect GPS measurements until the estimate of the transformation between the global GPS reference frame and our local map reference frame converges. This convergence can be observed in Figure 2. Once we have an estimate of the GPS to map reference frame, we can begin to incorporate GPS measurements into our existing mapping system. Given enough \((x, y, z)\) measurements from GPS, estimating the map to GPS transform is a well-posed problem. The procedure for estimating this transform is shown in Figure 3.

\[ ||T_p - T_{p'}|| \]

Fig. 2. The incremental change in the position component of the GPS to map frame transform with respect to the number of GPS measurements.

D. GPS measurement processing

Multipath GPS errors are caused when the line-of-sight GPS signal is weaker than a reflected signal [19]. Most
approaches to reducing this type of error have focused on
better receiver designs [19] or on statistical analysis [7]. Our
approach makes use of vehicle odometry to validate GPS
measurements before they are incorporated into the map.

The noise model reported by the GPS receiver is based
upon the number of satellites used in the fix. GPS measure-
ments are first ignored if they have a large measurement
covariance and are therefore based upon too few satellites.
As the robot enters or exits a structure, GPS data is often
subject to dominant multipath signals before satellite fixes
are lost while the receiver latches on to reflected signals.
When reflected signals are used to compute the GPS fix, it
will exhibit bias and additional error which is not captured
in the noise model; therefore, this measurement should be
rejected.

This validation procedure has two components. Once a
candidate GPS measurement is received by the validation
procedure, the current odometric pose corresponding to this
GPS measurement is preserved. Let $G_n$ be the GPS fix
$(x, y, z)$ coordinate at the current measurement number $n$.
Likewise, let $O_n$ be the odometric estimate of the robot’s
$(x, y, z)$ coordinate at the same instant in time as the
GPS measurement was received. The validation procedure
maintains a history of the past three of these measurements,
denoted $(G_n, G_{n-1}, G_{n-2})$ and $(O_n, O_{n-1}, O_{n-2})$. The
validation procedure accepts a GPS measurement if:

\[
||G_n - G_{n-1}|| - ||O_n - O_{n-1}|| < 0.1m
\]

\[
||\theta_G^2 - \theta_O^1|| - ||\theta_O^2 - \theta_O^1|| < 0.1 * \theta_m
\]

where

\[
m = \max(||G_n - G_{n-1}||, ||O_n - O_{n-1}||)
\]

\[
\theta_G = \tan^{-1}\left(G_n^y - G_{n-2}^y, G_n^x - G_{n-2}^x\right)
\]

\[
\theta_O = \tan^{-1}\left(O_n^y - O_{n-2}^y, O_n^x - O_{n-2}^x\right)
\]

\[
\theta_m = \max(||\theta_G^2 - \theta_G^1||, ||\theta_O^2 - \theta_O^1||)
\]

Briefly, these conditions state that the GPS and odometry
must agree within 10% of each other in the overall displace-
ment length, as well as the amount of angular change. See
Figure 4(a) for a visual description of these quantities. This
filter is used successfully in Figure 4(b) to prevent poor GPS
measurements from being incorporated into the map as the
robot enters a tunnel.

IV. EXPERIMENTS

Experiments were performed with the iRobot PackBot
robot which was augmented with additional computing re-
sources and sensors. The robot is equipped with a Core i7
computer, a Velodyne 32E LiDAR, and a MicroStrain GX3-
25 IMU. Two of the robots used in these experiments are
shown in Figure 5. The robots are tele-operated in these
experiments; however, in other work we have built maps of
environments through autonomous exploration [14].

Experimental scenarios were selected to establish the
performance of our mapping system in a variety of GPS
operational regimes. These GPS operational regimes vary
from mostly unoccluded such as along a road, exploration of
deep underground tunnels which block GPS until the robot
emerges, and mixed indoor and outdoor trajectories where
a robot enters and exits buildings and follows trails in a
forest. These types of scenarios are representative of the
operational environment of a ground robot in a military or
Fig. 5. iRobot PackBots used in these experiments. These robots have been augmented with GPS, IMU, LiDAR sensors, and onboard computing. The robot in Figure 5(a) is equipped with a custom 3D LiDAR scanner composed of a servo and a Hokuyo UTM-30-LX 2D laser scanner. The robot in Figure 5(b) is equipped with a Velodyne 32E 3D LiDAR.

rescue operation. A final GPS operational regime is where a robot is unable to ever get a GPS fix, such as when it remains indoors for its entire operation. This scenario is equivalent to mapping without GPS measurements, please refer to our prior work for details on indoor mapping without GPS [17], [10], [20], [21], [11].

The first experiment is designed to establish the usefulness of the GPS plugin in an open setting which admits continuous GPS signal visibility. The trajectory for this experiment lies along a sidewalk next to the road between two buildings which are 1 km apart. The robot maintains a GPS fix during most of its operation. The trajectory is flanked by a forest on either side of the road which provides some interference; however, the mapper is still able to accurately track the robot’s trajectory. The resulting map can be seen in Figure 6. For this visualization, aerial imagery was automatically fetched and displayed with the map data superimposed on top using the global coordinate reference generated through the process of mapping with GPS measurements. The point cloud data from the robot has been rendered along the path. This path matches well with aerial imagery.

In the second experiment, the GPS operational regime is where the robot starts outside with a GPS fix and enters a complex of underground tunnels. The robot maps a large loop before exiting from the tunnels at another location. In this scenario, the mapping procedure is able to operate within the tunnels using 3D laser scan measurements; however, some significant error is accumulated. This accumulated error is caused by the robot’s traveling a long distance down smooth walled tunnels; these smooth walls do not provide sufficient longitudinal constraints for ICP to determine the robot’s motion accurately. Once the robot emerges from the tunnel, it is able to re-acquire a GPS fix and it incorporates GPS measurements when they are consistent with the robot’s odometry using the validation procedure described in Section III-D. The robot “closes the loop” and re-enters the first entrance, following its previous trajectory.

The third experimental regime is tested by evaluating the performance of the mapper when the robot’s trajectory carries it through a variety of degraded GPS fixes. The robot starts in the street with an unoccluded view of the sky and gets a good quality GPS lock. The robot then enters a forest and proceeds along a trail. While in the forest, the GPS lock is degraded and sometimes interrupted; however, the robot is still able to incorporate this information. The robot then enters the back door of a warehouse where GPS is blocked completely. When the robot re-emerges from the warehouse, a GPS lock is re-acquired and the robot drives out into the road again where it has an accurate fix. The resulting trajectory can be seen in Figure 8.

V. CONCLUSION

Using GPS for mapping on ground robots is useful to establish a global frame of reference for mission execution as well as to correct accumulated map error. Unfortunately,
ground robots suffer from difficulty in using this resource which is caused by being confined to the ground such as multipath and occlusion. We have presented techniques which address these types of signal degradation to enable a mapping system to successfully incorporate GPS measurements.

In a GPS denied environment, such as in the interior of a building or cave, other sensor modalities must be used to estimate the robot’s position in the environment. The OmniMapper system presented in this paper uses 3D laser scanners and ICP to build maps of these types of environments and estimate the robot’s trajectory. When the robot is operated in an environment where a GPS fix can be maintained, such as the middle of a road, the mapping system can incorporate this information to establish a global frame of reference and provide corrections to eliminate accumulated error. In a GPS degraded environment, such as is seen when approaching or exiting a building, a validation procedure which compares robot odometry to relative GPS measurements is used to remove measurements corrupted with multipath error. A pose-graph framework like OmniMapper, is ideal for operation in degraded GPS environments since carefully filtered GPS provides extra information that can only improve the accuracy of the mapping process.

We have presented experiments which establish the performance of a mobile robot system using these mapping techniques in a variety of scenarios which touch on these three GPS operational regimes.

ACKNOWLEDGMENT

The authors would like to thank Jason Gregory for his assistance in data collection.

REFERENCES


