Regular Article

Field Testing and Evaluation of Single-Receiver GPS Odometry for Use in Robotic Navigation

Benjamin Congram and Timothy D. Barfoot
Institute for Aerospace Studies, University of Toronto, Toronto, Canada

Abstract: Mobile robots rely on odometry to navigate in areas where localization fails. Visual odometry (VO), for instance, is a common solution for obtaining robust and consistent relative motion estimates of the vehicle frame. In contrast, Global Positioning System (GPS) measurements are typically used for absolute positioning and localization. However, when the constraint on absolute accuracy is relaxed, accurate relative position estimates can be found with one single-frequency GPS receiver by using time-differenced carrier phase (TDCP) measurements. In this paper, we implement and field test a single-receiver GPS odometry algorithm based on the existing theory of TDCP. We tailor our method for use on an unmanned ground vehicle (UGV) by incorporating proven robotics tools such as a vehicle motion model and robust cost functions. In the first half of our experiments, we evaluate our odometry on its own via a comparison with VO on the same test trajectories. After 4.3 km of testing, the results show our GPS odometry method has a 79% lower drift rate than a proven stereo VO method while maintaining a smooth error signal despite varying satellite availability. GPS odometry can also make robots more robust to catastrophic failures of their primary sensor when added to existing navigation pipelines. To prove this, we integrate our GPS odometry solution into Visual Teach and Repeat (VT&R), an established visual, path-following navigation framework. We perform further testing to show it can maintain accurate path following and prevent failures in challenging conditions including full camera dropouts. Code is available at https://github.com/utiasASRL/cpo.

Keywords: navigation, position estimation, terrestrial robotics

1. Introduction

Odometry is an important component of almost any mobile robotic navigation strategy; it takes many forms including visual, visual-inertial, lidar, and wheel odometry. All of these variations use different sensors to accomplish the common goal of estimating the vehicle's path or trajectory. In mapping, odometry allows local reconstruction of the environment and in localization, it is critical

Received: 30 September 2021; revised: 13 June 2022; accepted: 26 July 2022; published: 30 August 2022.

Correspondence: Benjamin Congram, Institute for Aerospace Studies, University of Toronto, Toronto, Canada, Email: ben.congram@robotics.utias.utoronto.ca

DOI: https://doi.org/10.55417/fr.2022057

to the success of autonomous navigation systems such as Visual Teach and Repeat (VT&R) (Furgale and Barfoot, 2010). In experience-based navigation (EBN) (Churchill and Newman, 2013) and multi-experience localization (MEL) (Paton et al., 2016), odometry is used to bound pose uncertainty in short sections (i.e., less than 50 m) where localization fails due to factors such as appearance change. If odometry drift becomes too large, the robot may not be able to navigate safely. This in turn causes a missed opportunity to improve the map. Better odometry allows a robot to dead-reckon for longer sections and therefore drive further successfully.

One idea to improve a robot's odometry is to consider other sensors. Single-frequency GPS receivers are now ubiquitous, coming standard in almost every smartphone. First operational in 1983 (Seeber, 2008), GPS allows an absolute positioning solution to be calculated anywhere on Earth with a clear view of the sky. Since then, other Global Navigation Satellite System (GNSS) constellations such as GLONASS, Galileo, and BeiDou have come online and may be used independently or in combination with GPS. GPS has become an important tool for robotic navigation. However, standard pseudorange GPS positioning does not have sufficient accuracy to bound vehicle travel within the envelope required for visual localization, which typically degenerates with decimetre-level lateral errors (Furgale and Barfoot, 2010). Utilizing other GPS observables over short windows of time can improve relative positioning.

In this paper, we develop the Carrier Phase Odometry (CPO) project that uses time-differenced carrier phase (TDCP) measurements as the basis of a full odometry solution for UGVs. CPO is released as an open-source ROS2 (Quigley et al., 2009) package with code available at `https://github.com/utiasASRL/cpo`. In Section 5.1, we analyze CPO's performance on a set of trajectories collected by a UGV over several days. We then compare the performance of this single-receiver GPS odometry with stereo visual odometry (VO) on the same set of test trajectories in Section 5.2. To the best of our knowledge, this is the first study comparing and contrasting TDCP-based navigation with VO. Our results show CPO is a worthy alternative to VO in outdoor applications.

CPO can also provide benefits when GPS does not serve as the main sensor on a robot. As low-cost GPS receivers become standard, effectively utilizing the measurements from a single receiver becomes important even for mobile robots that primarily rely on rich sensors such as cameras or lidar. Robots unable to visually localize due to outdoor appearance change could drive longer distances via dead-reckoning provided they have good odometry estimates, for example. A self-driving vehicle relying on camera images needs a method to safely pull to the side of the road should that camera be blocked by stray debris. To validate these assertions, we integrate CPO into a highly successful visual path-following algorithm—VT&R. In a series of experiments, we show that single-receiver GPS odometry can improve performance and prevent total failure under challenging conditions.

Section 2 of this paper summarizes relevant work on GPS positioning and TDCP to provide background and put the remaining sections in context. In Section 3, we describe the theory used to develop our GPS odometry and apply it in VT&R. Our field-testing involves a total of five experiments and the procedures for these are explained in Section 4. In Section 5, the results are summarized and discussed. Finally, we conclude and suggest several directions for future work in Section 6.

2. Related Work

2.1. GPS positioning

Standard GPS positioning involves the trilateration of pseudorange measurements of four or more dedicated positioning satellites. In good conditions, pseudorange-based positioning can achieve accuracy on the order of 1–2 m despite several sources of error affecting the signal's measured time-of-flight (Kaplan and Hegarty, 2005). In addition to the pseudorange, GPS receivers can also calculate the carrier phase of the signal based on its Doppler shift (Seeber, 2008). These measurements are much less noisy but cannot be used directly due to the unknown integer ambiguity

of wavelengths between the receiver and each satellite. Real-time kinematic (RTK) processing uses a second, static receiver nearby. By comparing the pseudorange and carrier phase measurements of both receivers, the integer ambiguities can be resolved over time providing centimetre-level accuracy. However, the requirement for a dedicated base station and communications link make RTK expensive and impractical for many applications.

GPS positioning has become a useful tool for robotic navigation. A common strategy is to combine GPS measurements with those from an inertial measurement unit (IMU) often using a Kalman filter (Cooper and Durrant-Whyte, 1994) or factor graph optimization (Wen et al., 2019). The two sensors have several complementary properties. Together they may be used to detect low-frequency faults in the IMU measurements and high-frequency faults in the GPS measurements (Sukkarieh et al., 1999). More involved GPS techniques such as differential GPS (Zhao et al., 2014) and RTK (Scherzinger, 2006) have also benefited from fusion with IMU measurements. In this work, we investigate the potential of a single GPS receiver to provide accurate odometry estimates and avoid the requirement of integrating an additional sensor such as an IMU into the robotic platform. It should be noted, however, that our method is not incompatible with IMU measurements. Because of the flexible, factor-graph representation, these could potentially be added to further increase performance.

Sensors such as lidar and cameras have also been successfully combined with both low-cost GPS (Imperoli et al., 2018) and more advanced setups (Ohno et al., 2004). Typically, GPS is used as a global measurement to bound drift (Yu et al., 2019) in these sensor fusion algorithms. More recently, the use of other GPS observables such as Doppler velocity (Wen and Hsu, 2021) have been utilized. This can help improve the smoothness of the state estimate, especially during unstable GPS conditions (Cao et al., 2022).

2.2. Time-Differenced Carrier Phase

The idea of comparing carrier phase measurements from the same GPS receiver at different times was first proposed by Ulmer et al. (1995) but has received comparatively little attention in the robotics community. The technique, commonly known as TDCP, was developed for static geomatic surveying (Ulmer et al., 1995), (Michaud and Santerre, 2001), (Balard et al., 2006) but may be extended to full trajectories. When a receiver is in phase lock with a satellite, the ambiguity affecting carrier phase measurements is time-invariant. Within this period, differencing two phase measurements will cancel the ambiguity and avoid the need to resolve it. Therefore better accuracy can be achieved in estimating the relative receiver displacement between the two times, though absolute positioning accuracy remains high (Michaud and Santerre, 2001). TDCP has been used in applications as wide-ranging as vehicle convoying (Travis, 2010), (Pierce et al., 2017) and bird-flight trajectory reconstruction (Traugott, 2011). Success has been shown in combining TDCP with IMU measurements typically using a Kalman filter (Wendel and Trommer, 2004), (Wendel et al., 2006), (Zhao, 2016) or an iterated extended Kalman filter (Soon et al., 2008).

Doppler velocity integration is another technique that has been used for single-receiver positioning. The Doppler velocity observable is closely related to the carrier phase observable but the latter should be preferred for position estimation as it is less noisy (Wendel et al., 2003). Figure 1 compares the performance of the three single-receiver positioning techniques we have discussed over a set of short trajectories.

TDCP is versatile. In addition to relative positioning, TDCP-based techniques have been designed to solve for absolute position (Liu et al., 2013) and receiver velocity (Ding and Wang, 2011). The same "triple difference" idea that is key to TDCP can also be applied to differential GPS positioning (van Graas and Lee, 1995) and used for cycle slip correction (Kim and Langley, 2002). In this work, we optimize our implementation to solve for relative SE(3) pose—a typical output of robot odometry algorithms. Unlike previous TDCP work, we also incorporate several robotic state estimation tools including sliding-window filtering, robust cost functions, and a motion model compatible with typical robot kinematics.

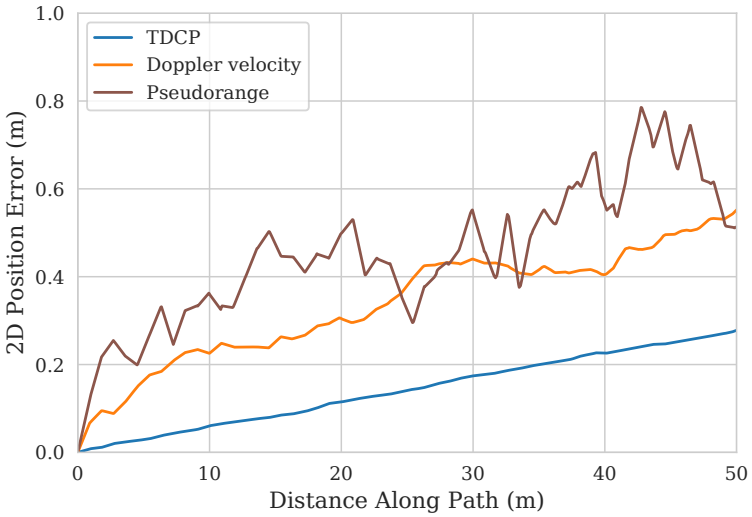

Figure 1. Comparison of the mean relative position error for three different techniques using a single GPS receiver across 12 independent paths. The TDCP method is both more accurate and much smoother than the pseudorange positioning while also outperforming the integrated Doppler velocity.

3. Methodology

We briefly summarize the coordinate frames relevant to the following sections. The global East-North-Up (ENU) frame, $\mathcal{F}_g$, is a stationary frame tangential to the Earth at the vehicle start position. All other frames are moving. The vehicle frame, $\mathcal{F}_v$, is located at the centre of the vehicle at axle height. All estimation is computed in $\mathcal{F}_v$ before being transformed to the GPS receiver frame, $\mathcal{F}_r$, for comparison with ground truth positions. The camera frame, $\mathcal{F}_c$, is located at the left camera of the stereo module. Finally, the origin of the satellite frame, $\mathcal{F}_s$, is defined at the antenna phase centre (APC) for calculating ranges. The VO algorithm is also configured to output estimates in the vehicle frame.

3.1. Carrier Phase Error Equation

Whereas RTK positioning makes use of carrier phase measurements from two receivers separated in space, TDCP positioning makes use of carrier phase measurements from a single receiver separated by both time and space. The carrier phase range equation to a single satellite at time a is given by

$$\Phi_a = \rho_a + N + c\delta_a^R - c\delta_a^S + E_a + T_a - I_a + m_a + \epsilon_a, \tag{1}$$

where Φ_a is the measured phase in radians multiplied by the known wavelength so that all values have units of metres. GPS receivers can measure the incoming phase quite accurately meaning the white noise affecting the measurement, ϵ, is typically less than 2 mm (Kaplan and Hegarty, 2005). However, the signal is affected by several sources of systematic error as it propagates from satellite to receiver causing the measured range, Φ, to differ from the true range to the satellite, ρ. These include receiver and satellite clock errors (δ^R and δ^S), satellite ephemeris error (E), tropospheric delay (T), ionospheric effects (I), and multipath (m).

N is the unknown wavelength ambiguity; if the receiver stays in phase lock with the satellite, N is time-invariant. Therefore, as shown in Michaud (2001), we can eliminate it by differencing (1) taken at two times, a and b:

$$\Phi_b - \Phi_a = \rho_{ba} + c\delta_{ba}^R - c\delta_{ba}^S + E_{ba} + T_{ba} - I_{ba} + m_{ba} + \epsilon_{ba}. \tag{2}$$

The subscript ba denotes the difference between a quantity at time b and time a. The receiver clock error is typically large so it must be dealt with explicitly, either by estimating it or differencing

the equation again for two different satellites to eliminate it. The latter gives us our measurement model:

$$\Phi_{ba}^{21} = \rho_{ba}^{21} - c\delta_{ba}^{S,21} + E_{ba}^{21} + T_{ba}^{21} - I_{ba}^{21} + m_{ba}^{21} + \epsilon_{ba}^{21}. \tag{3}$$

The term ρ_{ba}^{21}, for example, denotes the double-difference $\left(\rho_b^2 - \rho_a^2\right) - \left(\rho_b^1 - \rho_a^1\right)$ for a pair of satellites *1* and *2* at times a and b. The ranges making up ρ_{ba}^{21} are calculated using

$$\rho_a = \left\|\mathbf{r}_g^{sr}(t_a)\right\| = \left\|\mathbf{r}_g^{sg}(t_a) - \mathbf{r}_g^{rg}(t_a)\right\|, \tag{4}$$

where $\mathbf{r}_g^{sg}$ is the known satellite ephemeris and $\mathbf{r}_g^{rg}$ is our state. The notation $\|\mathbf{r}\|$ denotes the Euclidean norm of the vector $\mathbf{r}$. It is important to recalculate the ephemeris at each measurement time because the satellites travel at 3.9 km/s.

We will set up our estimation problem as an optimization with several factors (squared-error terms). From (3), we can write our error term for one pair of satellites seen at one pair of positions as

$$e_{ba}^{21} = \Phi_{ba}^{21} - \rho_{ba}^{21}. \tag{5}$$

Given n commonly seen satellites between t_a and t_b, our TDCP weighted least-squares factor is

$$J_{ba} = \sum_{k=2}^{n} w_k \left(e_{ba}^{k1}\right)^2, \tag{6}$$

where w_k is a scalar variance parameter, which we set as a constant in our implementation (though it could be tuned if more information on the measurement quality from each satellite was known). J_{ba} is symbolized as a blue dot in Figure 2. We can then optimize the least-squares factor given in (6) for our states, $\mathbf{r}_g^{rg}(t_a)$ and $\mathbf{r}_g^{rg}(t_b)$.

For optimization, a linearized error term is needed. We derive this by noting that $\mathbf{r}_g^{sr}(t_a)$ and $\mathbf{r}_g^{sr}(t_b)$, the vectors from the receiver to a particular satellite at t_a and t_b, are approximately parallel for small t_{ba} since the distance between receiver and satellite is much larger than the distance either travels in this timespan. As illustrated in Figure 3, the range to the satellite can change due to both the receiver's movement and the satellite's movement between measurement times. For succinctness, we define the unit vector from the receiver to the satellite as $\mathbf{u} = \frac{\mathbf{r}_g^{sr}(t_a)}{\left\|\mathbf{r}_g^{sr}(t_a)\right\|}$. From Figure 3 we see that

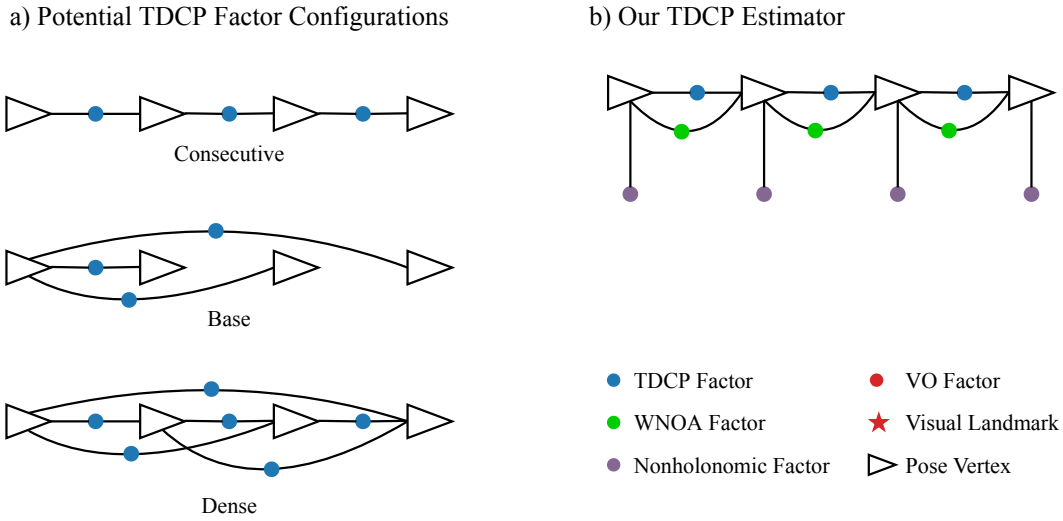

Figure 2. (a) Potential ways TDCP factors can be added. Due to the error characteristics of the phase range, they all give very similar position estimates. The consecutive configuration was chosen for our estimator. (b) Factor graph for our TDCP algorithm.

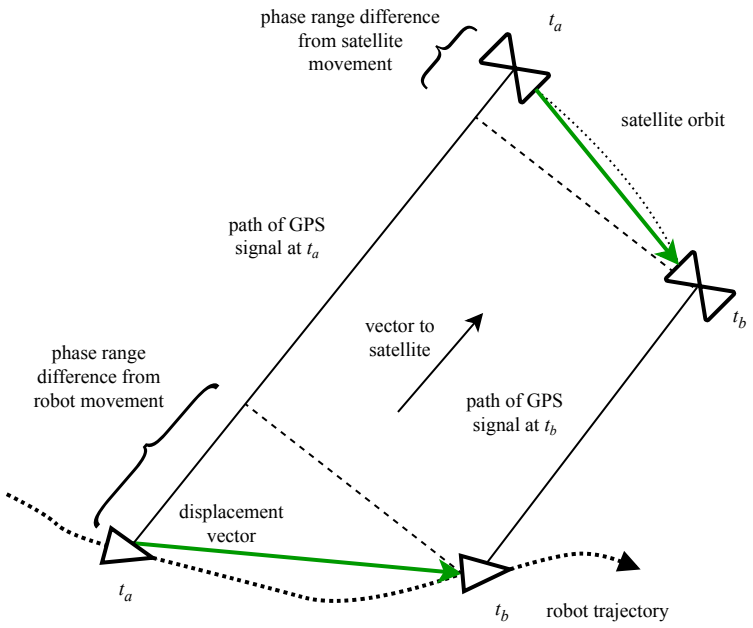

Figure 3. In linearizing the error term, we make the assumption that the unit vectors from receiver to satellite at times *a* and *b* are parallel. As a result, the difference in measured range due to receiver movement has the same magnitude as the scalar projection of the receiver displacement vector onto the satellite vector.

the range difference due to the satellite's cross-track movement is equal to the satellite displacement vector projected onto this unit vector. Likewise, the change due to receiver movement (i.e., the robot driving) corresponds to the negative of the receiver displacement vector projected onto $\mathbf{u}$. Combining these gives:

$$\rho_{ba} = -\hat{\mathbf{u}}^T \left(\mathbf{r}_g^{rg}(t_b) - \mathbf{r}_g^{rg}(t_a)\right) + \hat{\mathbf{u}}^T \left(\mathbf{r}_g^{sg}(t_b) - \mathbf{r}_g^{sg}(t_a)\right), \tag{7}$$

where the second half of the right-hand side (the satellite movement term) is independent of the state. After substituting (7) into our error equation, (5), we can calculate the Jacobian required to perform Gauss-Newton optimization. Details on the implementation of this optimization are given in Section 3.4.

3.2. Carrier Phase Noise Properties

Our error equation, (5), constrains the transformation between the receiver pose at two times. Given a set of carrier phase measurements collected at a fixed rate (e.g., 1Hz), we have a choice of how to pair these measurements to form error terms. Figure 2(a) illustrates three potential options. If our measurements at each timestamp were primarily affected by Gaussian noise, then the dense strategy would be best. More factors in our factor graph would average out the noise and improve our estimates. However, looking at our measurement model, (1), we see that the majority of error sources are systematic in nature. That is, if the receiver could take two measurements of the carrier phase at the same instant, they would be almost exactly the same save for the very small measurement error, ϵ. The other error sources vary smoothly. Neglecting ϵ, we would find that e_{ca}, the error term on the poses at times a and c, is a simple linear combination of e_{cb} and e_{ba}. Therefore the additional error terms in the dense configuration compared to the other configurations add very little to the optimization problem besides computational burden. The base and consecutive strategies are very similar, but the consecutive method has subtle advantages when the set of satellites available is time-varying. In base, neighbouring pose estimates with respect to the base vertex may be calculated with different satellites so the transformation between these vertices is liable to be less smooth.

Some of the errors in (1), the phase range equation, can be mitigated through modelling. It is typical to use the Klobuchar model (Klobuchar, 1987) to partially correct for ionospheric effects, the parameters of which are available in the GPS navigation message. The Niell mapping function (Niell, 1996) with the UNB3 model parameters (Collins et al., 1996) can be used to estimate the tropospheric delay. Both models are a function of atmospheric conditions and satellite elevation. Because atmospheric conditions change slowly and the errors are differenced in (7), their impact is lessened compared to the effect on a single phase measurement. However, the effect of satellite elevation change over the time difference can be significant for satellites close to the horizon. In our experiments, we model the tropospheric delay but omit the ionospheric correction because the applicable messages were not logged for all runs. The impact of atmospheric delays is further explored in Section 5.1.1.

3.3. Carrier Phase Cycle Slip

A key requirement for being able to use our TDCP error equation is that the satellites involved must have maintained continuous phase lock over the interval. In practice, this is not always guaranteed as cycle slips can occur due to factors such as satellite occlusion, low signal-to-noise ratio, or the multipath effect (Dai et al., 2009). Several cycle-slip detection methods exist and the optimal choice depends on the type of GPS receiver being used. For multi-frequency receivers, the carrier phases at two frequencies may be differenced to eliminate the majority of error sources and the continuity of the result may be used as a test signal (Subirana et al., 2013). If two or more GPS frequencies are available for the particular receiver, our method will take advantage of this approach. If only one frequency is available, cycle slip can still be detected, albeit in a less robust manner, by comparing the pseudorange and carrier phase measurements over time (Subirana et al., 2013).

When a cycle slip occurs, the satellite it affects is excluded from our optimization until continuous phase lock is reacquired. The remaining satellites can still be used to form TDCP cost factors. As detailed in Section 3.4, a robust cost function is applied for each term in the least-squares optimization problem. As a result, outlier measurements—those that strongly disagree with the other measurements and our motion model—will have minimal influence on the optimization result. This means that even if our detection method fails to catch a cycle slip, our final state estimate will not be significantly biased.

3.4. Odometry Implementation Details

Using the theory from previous sections, our goal was to create an easy-to-use package that would provide live GPS odometry estimates to be used alone or as part of a larger navigation stack. The result is the open-source CPO project found at `https://github.com/utiasASRL/cpo`. CPO is a Robot Operating System (ROS)2 project designed to work with the majority of modern GPS receivers. It consists of four packages. The `cpo_frontend` package acts as a driver and preprocessor for the carrier phase measurements. The input is standard RTCM1004 (GPS observables) and RTCM1019 (GPS ephemerides) messages logged over serial. The output of this package is a stream of custom TDCP messages, defined in the `cpo_interface` package, published to a ROS2 topic. These messages act as pseudomeasurements pairing a set of satellites observed at two consecutive time points. The front-end node parses the binary RTCM messages, calculates approximate pseudorange GPS solutions, and extracts vectors relevant to the estimation problem. It can detect when the receiver loses phase lock with a particular satellite between consecutive GPS measurements and exclude that satellite until phase lock is regained. This ensures the integer ambiguity is time-invariant for all measurement pairs we use. Typically, phase lock will be maintained with the other satellites such that a high-quality odometry estimate can still be obtained. The front-end node also estimates and corrects for the tropospheric delay difference as discussed in 3.2.

The `cpo_backend` package is responsible for state estimation. Each pseudomeasurement message received from the front end is used to construct $n-1$ TDCP error terms, where n is the number

of observed satellites. Carrier-phase measurements may be subject to outliers so a robust cost function [dynamic covariance scaling (DCS) (Agarwal et al., 2013)] is used on the TDCP factors. Our algorithm is designed and tested for a nonholonomic robot so factors that penalize lateral velocity of the vehicle frame are added to the nonlinear least-squares cost function. We also use a white-noise-on-acceleration (WNOA) motion prior (Anderson and Barfoot, 2015) to encourage smoothness. Together, these allow us to resolve the vehicle orientation. Yaw and pitch are determined by these factors implicitly aligning the longitudinal axis of the robot with the direction of the vehicle frame velocity while roll is determined by the shape of the trajectory. To fit our use case, we have assumed a ground vehicle on a primarily planar surface. Unlike other TDCP algorithms, the use of a motion model such as this allows the robot to make use of carrier phase information and still calculate a state estimate when less than four phase-locked satellites are available.

Optimization is handled by the simultaneous trajectory estimation and mapping (STEAM) library (Anderson and Barfoot, 2015) over a sliding window. The user may use ROS2 parameters to easily configure the relative importance of these factors as well as the size of the sliding window to fit their application. The result is full SE(3) pose estimates of the vehicle in the ENU frame. These are published with a standard ROS2 `PoseWithCovariance` message either at a fixed rate or with each new incoming pseudomeasurement. This node also provides a query trajectory service. The service accepts two timestamps and returns the relative pose and its covariance over the interval. The two query times do not need to be at GPS measurement times as the continuous-time trajectory can be queried at any point (Barfoot et al., 2014).

3.5. GPS Odometry in VT&R

CPO was designed to be flexible such that it may be used as a standalone package or integrated into a more complex navigation stack. To validate this, we will show how our odometry method can be added to VT&R to increase the autonomy rate of an already successful visual path following system. We will first define VT&R and briefly describe its existing operation before explaining how our GPS odometry may be utilized within its framework.

Autonomous path following through highly unstructured environments is a challenging yet important task for mobile robots. VT&R utilizes a single, manually driven training example to achieve this goal using only one stereo camera (Furgale and Barfoot, 2010). Similarly to simultaneous localization and mapping (SLAM), the robot performs VO to estimate its relative motion and visual localization to estimate its place in the map. Because it is repeating the same path, VT&R is able to leverage the deliberately consistent camera viewpoint. The result is an extremely high autonomy rate combined with few-centimetre-level path-tracking accuracy.

The success of VT&R is due in large part to the locally metric, globally topological map formulation. Its pose graph stores measurements and transformation estimates in relation to their neighbours rather than in a single privileged coordinate frame. This allows for state estimation that is both highly accurate and computationally efficient. Computational complexity is decoupled from map size. The formulation can be seen as an extension of submapping techniques (Chong and Kleeman, 1999), (Williams, 2001), (Marshall et al., 2008) and the relative bundle adjustment method of Sibley et al. (2009). Another key to VT&R's success is the interleaving of dead-reckoning from VO and localization (Furgale and Barfoot, 2010). We will show how we may use our GPS odometry as an alternative form of dead-reckoning in the prediction step of VT&R.

We explored two possible strategies for integrating our GPS odometry into VT&R. Our first approach was to add TDCP terms into the sliding-window bundle adjustment stage of its odometry pipeline. In this tightly-coupled setup, the front-end node of CPO would run, publishing ROS2 messages that contain all the information needed to construct a TDCP factor. VT&R could then subscribe to these messages and easily accommodate these new factors in the existing nonlinear least-squares optimization problem.

Our second approach recognizes that accurate odometry is not a direct requirement for VT&R and the path-tracking problem. Rather, odometry is utilized as a prior for the localization problem.

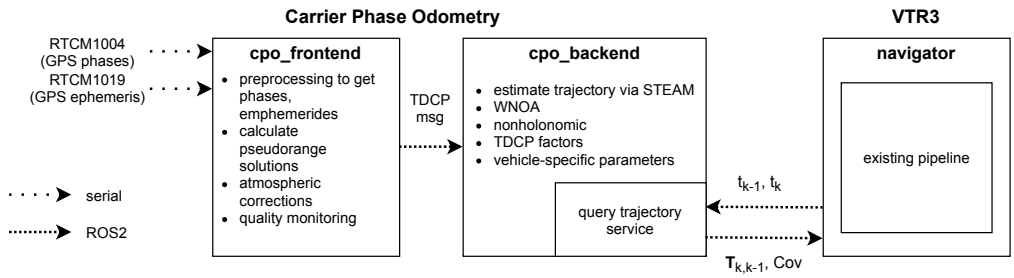

Figure 4. Architecture diagram for our chosen approach to include GPS odometry in VT&R. All GPS estimation is done separately from VT&R which can query the estimator to receive transformation estimates for its intra-run edges.

In this 'lazy' approach, we calculate odometry using both sensors (when available) independently in separate threads. GPS odometry is computed in the `cpo_backend` node while VO is computed in VT&R's navigator node as normal. We then let the localizer decide which of the two odometry solutions is the best source to use as its prior when the time comes, instead of fusing the two solutions directly. A diagram of the setup can be seen in Figure 4.

There are several advantages the "lazy" approach has over the tightly-coupled solution. Perhaps the largest is that, from a practical point of view, it is much more easily extended to versions of VT&R that swap the stereo camera used in this work for other primary sensors such as lidar or radar. The tightly-coupled strategy required careful handling of edge cases in the stereo bundle adjustment pipeline and the implementation would need to be rewritten for new bundle adjustment pipelines. In the lazy strategy, the bundle adjustment pipeline does not need to know that GPS exists. Another advantage of the lazy approach is that we do not have to balance the size of our sliding window between the two sensors. We are free to optimize the sparse GPS factors over a larger window while keeping the visual bundle adjustment over a small window containing many stereo landmark factors. As well, parameter tuning becomes less important when we are not fusing the two sensors directly. VT&R can get away with unrealistic covariances on its stereo landmark terms to some degree because only the relative weighting of terms is important. When adding GPS terms to the optimization problem, suddenly the results become more dependent on our choice of parameters. Finally, the lazy approach avoids having to estimate and bookkeep the global orientation in VT&R. In the tightly-coupled approach, the global orientation becomes an extra state variable that must be included in our state vector if and only if we are using GPS factors. In the lazy approach, it is encapsulated in CPO.

In theory, the careful combination of two reliable measurement sources in the tightly-coupled approach should produce a probabilistic estimator that outperforms either source alone. However, even after careful tuning of parameters in the tightly coupled algorithm, we could not produce an estimator that consistently outperformed both single-sensor odometry algorithms because of these practical disadvantages. The 'lazy' strategy avoids these issues and as a result is used in the experiments presented in this paper. The remainder of this section provides details on the implementation.

Because of our chosen approach, our algorithm runs very similarly to VT&R without GPS with a couple of key differences. After a new keyframe is created and bundle adjustment has run, a request is sent to CPO's query trajectory service with the timestamp of the current keyframe, t_k and the previous keyframe, t_{k-1}. The response, $\mathbf{T}_{k,k-1}$ and its covariance, is stored in the pose graph edge as a second, separate transformation. As the current keyframe was just captured and the GPS measurements are asynchronous with respect to the camera images, the query trajectory service typically has to extrapolate slightly past the latest GPS measurement. However, this is easily handled by STEAM's continuous-time estimation. To obtain a better estimate of $\mathbf{T}_{k,k-1}$ utilizing GPS measurements received after t_k, the service is called again after a fixed delay and the VT&R pose graph is updated accordingly.

The new transformations are depicted as green edges in Figure 5. In the localization problem, we are required to estimate the transformation between the current repeat vertex, V_d, and the closest

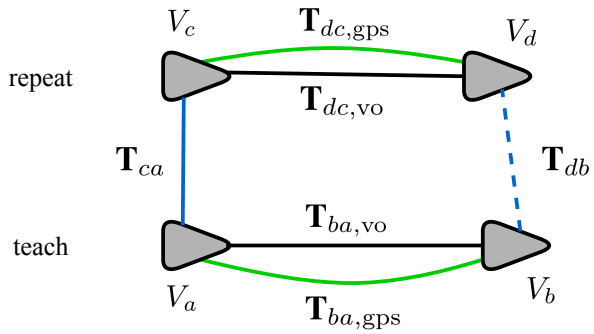

Figure 5. Diagram of the localization chain showing the part of the pose graph relevant to the localization problem. Transformation estimates from odometry within each run may be composed with the previous localization result, $\mathbf{T}_{ca}$, to provide a prior for the current estimation problem.

teach vertex, V_b (Figure 5). The standard version of VT&R composes the edges from VO with the most recent localization result to generate a prior:

$$\check{\mathbf{T}}_{db,\text{vo}} = \mathbf{T}_{dc,\text{vo}}\mathbf{T}_{ca}\mathbf{T}_{ba,\text{vo}}^{-1}. \tag{8}$$

With GPS odometry available, we now have an alternative method for generating this prior:

$$\check{\mathbf{T}}_{db,\text{gps}} = \mathbf{T}_{dc,\text{gps}}\mathbf{T}_{ca}\mathbf{T}_{ba,\text{gps}}^{-1}. \tag{9}$$

In the event of VO failure, we can now still calculate a prior via the GPS edges. If no GPS is available, we can still use VO as before. If both are available, we can either compare their covariances to determine which is likely to be more accurate or the user may decide to always prefer one sensor or the other.

One reason a good prior is important is that it provides a good initial condition for localization. Perhaps a more significant reason is that in the case where localization fails, the prior becomes our estimate for $\mathbf{T}_{db}$. When VO fails, visual localization is also likely to fail so the GPS odometry prior becomes very important.

We note this is not the only way GPS could be used in VT&R; Congram and Barfoot (2021) provide an alternative way to fuse vision and GPS for localization in a way that does not require a privileged frame or any post-processing of the map. However, our method in this paper has the advantage of only requiring a single GPS receiver.

4. Experimental Setup

All data were collected aboard the Clearpath Grizzly UGV pictured in Figure 6. The vehicle maintained an average velocity of 1m/s across terrain that included pavement and snow-covered grass at the University of Toronto Institute for Aerospace Studies (UTIAS) campus. Stereo images were captured by a front-facing Point Grey Research Bumblebee XB3 stereo camera, which has a 24 cm baseline, a 66° horizontal field of view and captures 512 × 384 pixel images at a 16 Hz framerate. GPS measurements were recorded by a NovAtel SMART6-L receiver mounted near the front of the vehicle. Carrier phase measurements were logged at 1 Hz while RTK ground truth was logged separately at 4Hz. The RTK positioning is expected to have an RMS error of 1cm + 1ppm under nominal conditions.

4.1. GPS Odometry Experiments

The goal of the first experiment was to analyze the performance characteristics of our GPS odometry algorithm. To collect the data, the Grizzly drove 11 separate runs over three data collection days while logging raw GPS observables. Two of these days were during winter and the other during summer. Figure 7 shows some sample images representative of the data collected. Each path featured

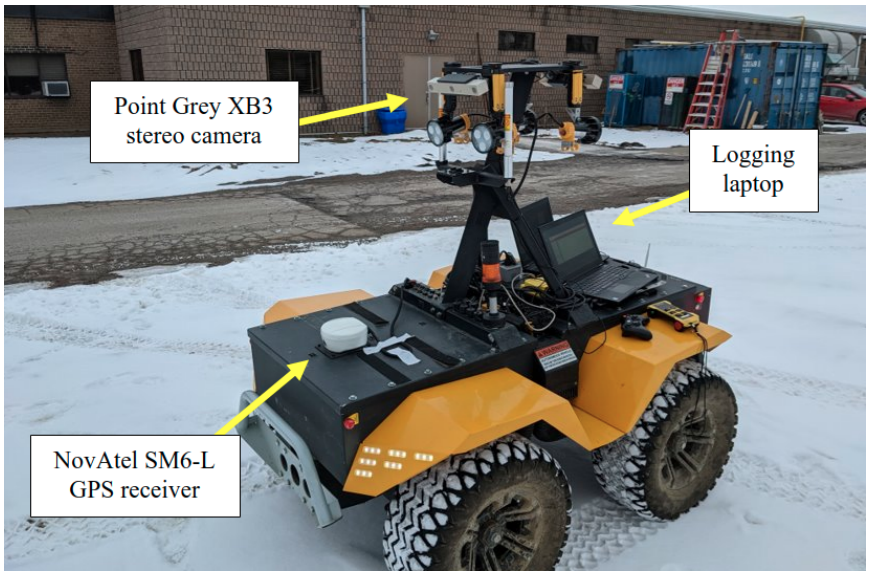

Figure 6. The Clearpath Grizzly Robotic Utility Vehicle used for data collection.

Figure 7. Examples of images from the dataset used in the experiments. The varied environment included snow, grass, and tall vegetation. Runs were primarily collected during daytime, although one path used in Section 5.3 was collected at nighttime.

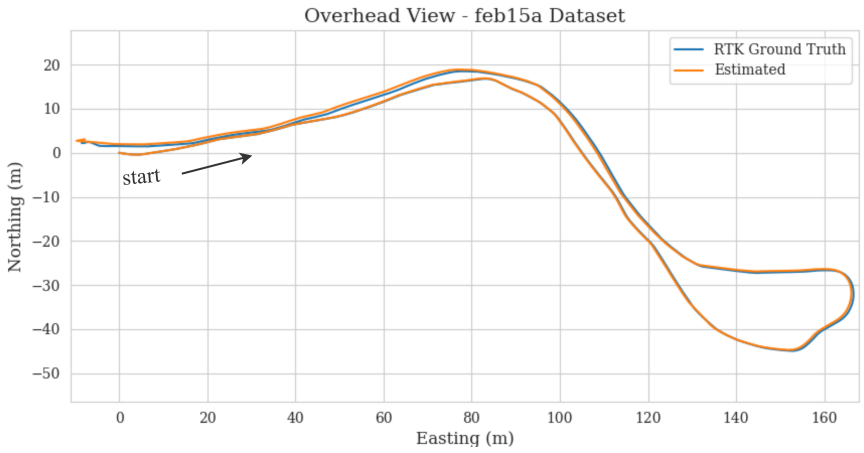

Figure 8. Overhead view of a representative run from the dataset showing the estimates of our GPS odometry algorithm compared to the RTK ground truth. The estimates are smooth and accurate. The trajectory is almost indistinguishable from the ground truth at this scale.

several turns of varying radii. An example of one of these paths can be seen in Figure 8. Paths ranged in length from 209 to 404 m.

In this experiment, the data from each run were played back in real-time while the CPO algorithm ran using its default parameters. The start of each of the trajectory estimates was placed coincident to the ground truth start point allowing our estimates to be compared with ground truth for the rest of the path. These results are presented in Figure 9.

We also perform two smaller studies. The first evaluates the impact of utilizing the tropospheric correction in our estimator. The second validates that our estimator can still make use of the carrier phase measurements when less than four phase-locked satellites are available. Results for these two experiments are provided in Sections 5.1.1 and 5.1.2, respectively.

4.2. Comparison to Visual Odometry

The second experiment involved a study comparing TDCP odometry to VO to provide a point of reference and a further evaluation. The stereo VO method used is based on parallel tracking and mapping (PTAM) (Klein and Murray, 2007) and is the same as is used in VT&R. Motion estimates are computed at framerate while landmark positions are optimized in a windowed bundle adjustment after each keyframe. It is fast and reliable with the parameters pre-tuned for use on the Grizzly UGV used to collect the data for our experiments.

On 8 of the 11 runs collected, stereo images were also logged. These runs were then split into a total of 24 independent 50 m sections, approximately equally spaced, for evaluation. We chose 50m as an evaluation distance as we do not anticipate driving a robot on dead reckoning farther than this and it is sufficient for measuring odometry drift rate. As VO does not estimate orientation in the global ENU frame, the 10 m of trajectory preceding the test section was used for alignment of the VO estimates. The continuous-time trajectories computed by STEAM are used to interpolate the VO estimates to the ground truth GPS timestamps (as they are asynchronous to the VO keyframe timestamps). Evaluation is considered based on the amount of drift (absolute planar translation error) after 50 m. These results can be seen in Figure 12.

4.3. Visual Teach and Repeat Experiments

Three further experiments were performed to study the effectiveness of using GPS odometry in VT&R as discussed in Section 3.5. They were designed such that each one could validate a goal for our combined algorithm.

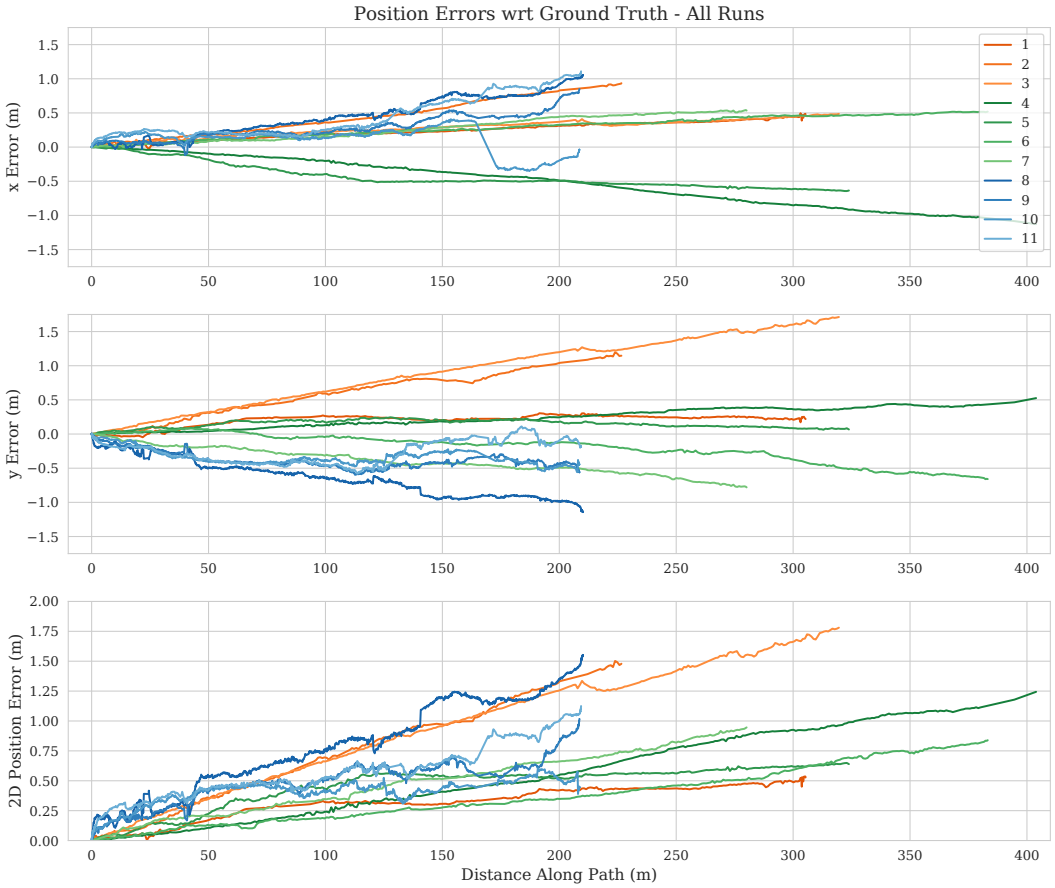

Figure 9. GPS odometry estimate errors with respect to ground truth for each test run. Runs are coloured based on the day they were collected. Orange runs were collected on February 10, green runs on February 15, and blue runs on August 6. While performance varies by run, all trajectories have a final positional error of less than 2m.

1. Replacing the VO prior used by Visual Teach and Repeat 3 (VTR3) localization with a GPS odometry prior provides comparable performance under nominal conditions. In theory, the more accurate GPS odometry could provide a better prior than VO and therefore more accurate localization. However, VO has shown it provides a sufficient prior for this task in normal circumstances, so we only need to show that GPS can do at least as well.
2. The GPS prior can be relied on for dead-reckoning when we are not able to localize visually. VTR3 often faces small sections where it cannot visually localize due to factors such as appearance change. To test this we will create a difficult scenario for the robot by not allowing the localization pipeline to use any feature matches for certain sections.
3. GPS odometry can be used to maintain path-following in the event of total VO failure. We will test this by replacing the left image from the stereo camera with a blank image simulating, for example, a total blockage of the camera. With GPS odometry, VTR3 should be able to regain normal localization once the camera recovers. We will also show that without GPS, this would result in a failed repeat.

We run each experiment first using a version of VTR3 without GPS, and then using our method configured to replace VO with GPS odometry. We test on seven independent path combinations. The first three paths are unique and approximately 60 m in length (one minute of driving per run).

They are driven twice with one run serving as a teach and the second a repeat. For experiments 2 and 3, the sensor dropout is applied to a 20-second section midway through the repeat run to illustrate the behaviour both when the dropout begins and when the sensor recovers. The terrain seen during paths 1–3 is a combination of pavement, maintained grass, and small buildings. Paths 4–7 originate from several traversals of a path approximately 210 m in length. A longer dropout encompassing approximately 100 m of driving is applied for repeats on these paths to test more extreme scenarios. These paths were driven through a more natural environment with tall grass and trees having been seen. The repeat of path 1 was manually driven with a large path-tracking error to intentionally create challenging conditions. All other paths were autonomously driven providing conditions more commonly experienced in VT&R. The results of these experiments are provided in Section 5.3.

5. Results

5.1. Single-Receiver GPS Odometry

Satellite availability for the GPS-only experiment varied throughout the runs as buildings and even the vehicle sensor mast itself caused partial occlusions of the sky. Despite this, the receiver kept enough satellites in phase lock throughout the runs for a consistent position estimate at all times. The median number of satellites seen was seven with a minimum of four and a maximum of nine. Raw GPS data were logged for the first seven runs from which dilution of precision information could be extracted. The minimum horizontal dilution of precision (HDOP) seen during these runs was 0.9, the maximum HDOP was 4.4, and the median HDOP was 1.3.

Figure 8 shows the overhead view of our estimates versus the ground truth for one representative run in the dataset. From this plot, we see the estimates closely match the shape of the ground truth path and are smooth throughout. Table 1 summarizes the results for all 11 paths in the dataset over 200 m in length. These runs each encompass an average of five minutes of driving, providing enough data to characterize the error growth properties while dead-reckoning as seen in Figure 9. We calculate drift rate for a given run as the horizontal position error with respect to ground truth divided by distance travelled. We find the average drift rate across all runs, weighted by path length, to be 0.38%. The worst-case run, run 8, still has a final drift of less than 1% of the trajectory's length. Together, these runs constitute 3.1km of driving.

As seen in Figure 9, errors from different data collection days are of similar magnitude to each other, despite the different satellite conditions. Interestingly, the runs collected in August (blue lines in Figure 9) are less smooth than the other runs, likely because the GPS measurements were collected at a higher frequency leading to higher frequency estimate updates (10 Hz versus 1 Hz). This may also be a result of the varying satellite availability as the robot drove through the wooded area. In the February runs, the errors grew smoothly and approximately linearly. Near the end of

Table 1. GPS odometry results for full runs

Dataset	Length (m)	Final Error (m)	Drift Rate (%)
1	305	0.53	0.17
2	227	1.48	0.65
3	320	1.78	0.56
4	404	1.24	0.31
5	324	0.64	0.20
6	383	0.84	0.22
7	280	0.95	0.34
8	210	1.55	0.74
9	209	1.02	0.49
10	209	0.45	0.22
11	209	1.13	0.54

Table 2. Effect of correcting for tropospheric delay difference on final error

		Final 2D Translation Error (m)	
Dataset	**Distance Travelled (m)**	**Without Tropospheric Correction**	**With Correction**
4	404	1.66	**1.24**
5	324	0.66	**0.64**
6	383	1.34	**0.84**
7	280	0.98	**0.95**

Table 3. Performance given 50m section of limited satellites

		Final 2D Translation Error (m)	
Dataset	**Distance Travelled (m)**	**Full Satellites**	**Limited Satellites**
4	404	**1.24**	1.52
5	324	**0.64**	1.76
6	383	0.84	**0.76**
7	280	**0.95**	3.02

run 10, the surrounding foliage briefly prevents the estimator from using GPS measurements leading to a spike in the error. Importantly, the robot is able to fall back on its motion model during this section and the final dead-reckoning error is still reasonable.

5.1.1. Impact of Tropospheric Correction

There is some discrepancy among prior works that use TDCP on the importance of correcting for atmospheric delays. Some works (Traugott, 2011) correct for it while others (Suzuki, 2020) assert that over the small time period the carrier phase difference is calculated the delays can be considered constant. To test this empirically we also run our algorithm on the data without the tropospheric correction. Final errors are summarized in Table 2. We find on all runs the (relatively easy to calculate) correction improves the final error, thus supporting its use. Interestingly, the performance gain varies between runs, likely due to differences in satellite geometry. The tropospheric delay is roughly proportional to $1/\sin\epsilon$, where ϵ is the satellite elevation. Therefore the delays are more variable and the correction is more important when using measurements from satellites lower in the sky.

5.1.2. Performance with Limited Satellite Availability

In pseudorange-based positioning, a minimum of four satellites are required to resolve a receiver position. Using TDCP measurements alone, four phase-locked satellites are also needed to calculate a relative baseline vector. In areas where less than four satellites are available, these algorithms must either reset their estimation or use another method to bridge the gap such as the noisier pseudorange solutions on either end of the section (Traugott, 2011). However, the use of a motion model in CPO means we can still make use of the carrier phase measurements when only two or three satellites are in phase lock. To show this, we perform an experiment on four of our datasets during which we artificially limit satellite availability for a 50 m section. For each run, the estimator is only given the first three satellites seen between 50 and 100 m along the path. The results are compared in Figure 10 to the estimates with full satellite availability. The final errors are summarized in Table 3.

Overall, the results are worse when satellite availability is limited in the 50m section, as expected, though on runs 4 and 6 the final errors are comparable. The largest difference is seen in Run 7. In this trial, the robot performed a series of significant turns during the section with limited availability and finished the section travelling in the opposite direction. Though the CPO solution drifts, the results are still much better than if we had relied on a motion model alone (effectively defaulting to a constant velocity prior). Using GPS alone, a real-time solution could not be determined in these sections for any of the runs. Instead, our algorithm is able to utilize the limited carrier phase

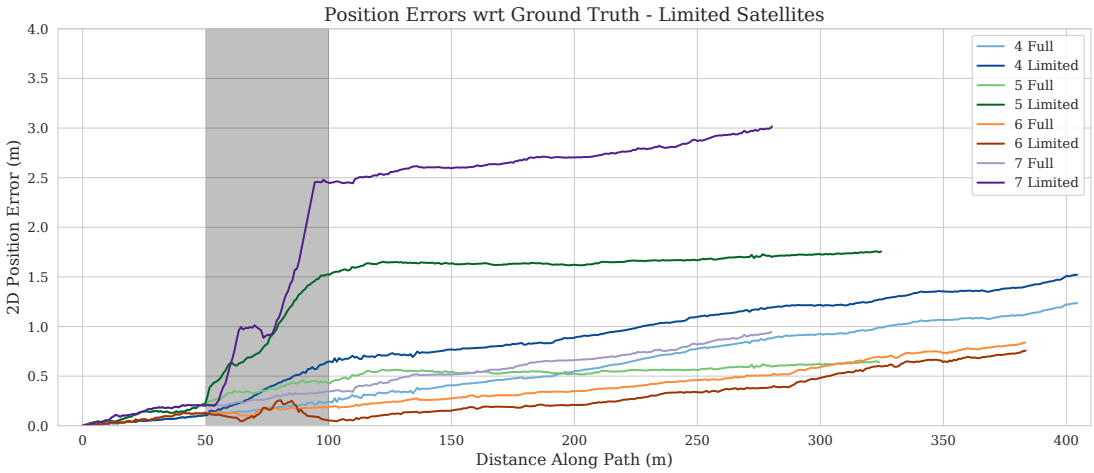

Figure 10. Comparison of errors on four different datasets during the limited satellite availability experiments. Between 50m and 100m along the path (grey background), the estimator was only allowed to use the first three satellites seen in the limited scenario (darker lines). As expected, performance is worse compared to when full satellites are available. However, our estimate is still able to make use of the GPS measurements and provide a reasonable state estimate.

information while relying on the motion model to resolve the remaining degrees of freedom. As a result, the average drift rate over these runs is only 0.51% in the partially limited satellite availability scenario.

5.2. Comparison to Visual Odometry

Figure 11 shows an overhead view of the estimates from both GPS odometry and VO on three of the test trajectories representative of the larger test set in the second experiment. Even at this macroscopic scale, we can see the GPS odometry outperforms VO. Figure 12 depicts both the errors for the individual runs and an average horizontal position error for each algorithm. After 50 m, the TDCP method has a smaller translational error than VO on all but two of the 24 test trajectories. VO has a mean final translational error of 1.02 m or 2.04% while TDCP does 79% better with a mean error of 0.21 m or 0.42%. The variance in drift rate between runs is also a lot higher for VO as can be seen in the spread of data in Figure 12. This implies the expected errors may be more predictable for TDCP.

A similar number of satellites were available for the comparison experiment as in the GPS-only experiment with the minimum five, the median seven, and the maximum nine. We note that the Grizzly's GPS receiver was not configured for use with a power-hungry stereo camera nearby so was somewhat affected by electromagnetic interference. Proper shielding may have improved the satellite availability. Looking more closely at the VO results, we see the number of feature matches varies somewhat between the two major types of terrain seen—dry pavement and snow, but is enough for a reasonable motion estimate throughout. There were no VO failures (i.e., there were always enough landmark matches to produce a consistent estimate). We notice the VO tends to slightly overestimate or underestimate distances within a run. As a result, the total translational error over a full loop trajectory is smaller than the drift rate for shorter sections though still significant. Further tuning might be able to improve VO performance slightly, but it is unlikely to reach the level of the GPS odometry. Finally, we note the TDCP method also has computational advantages as it only requires one error term per satellite pair compared to the potentially hundreds of stereo landmark terms involved in VO. In our head-to-head comparison, GPS odometry was clearly superior in a two-dimensional planar setting.

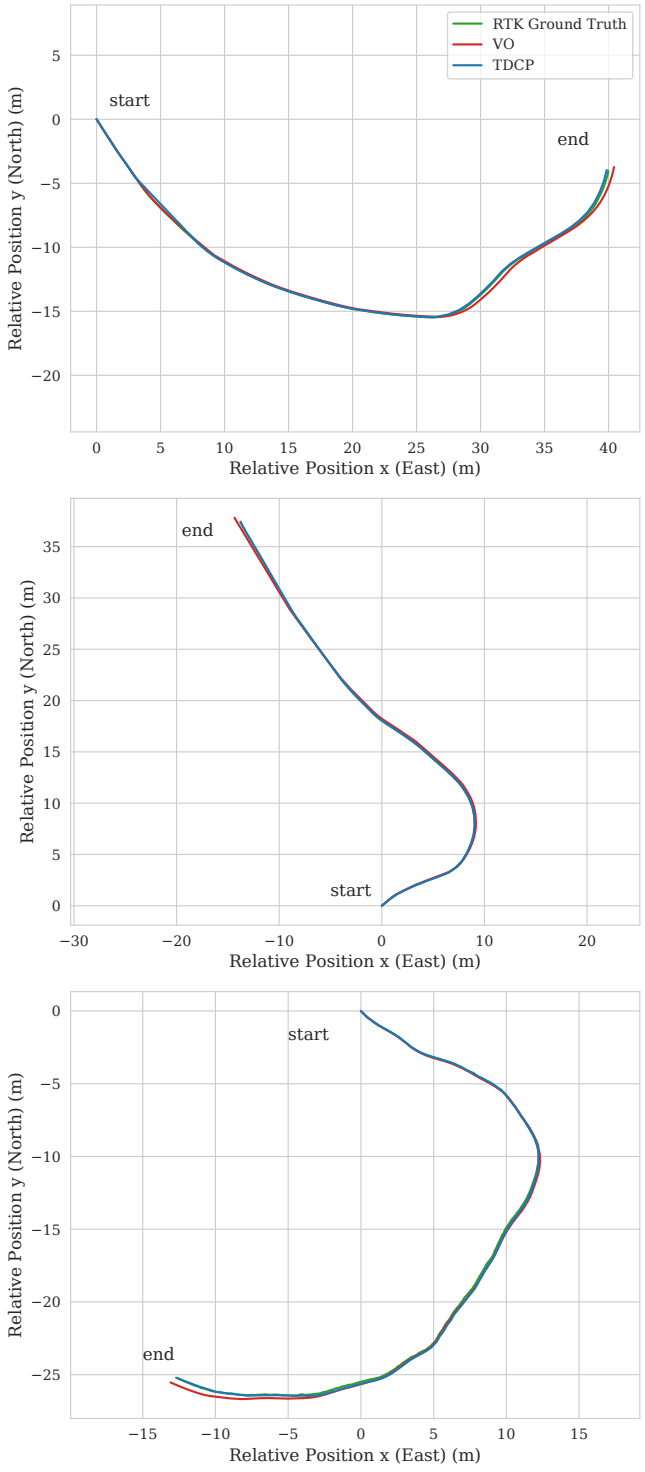

Figure 11. Overhead view of ground truth and estimates for three of the 24 test trajectories. VO drifts noticeably further from ground truth than the TDCP-based odometry.

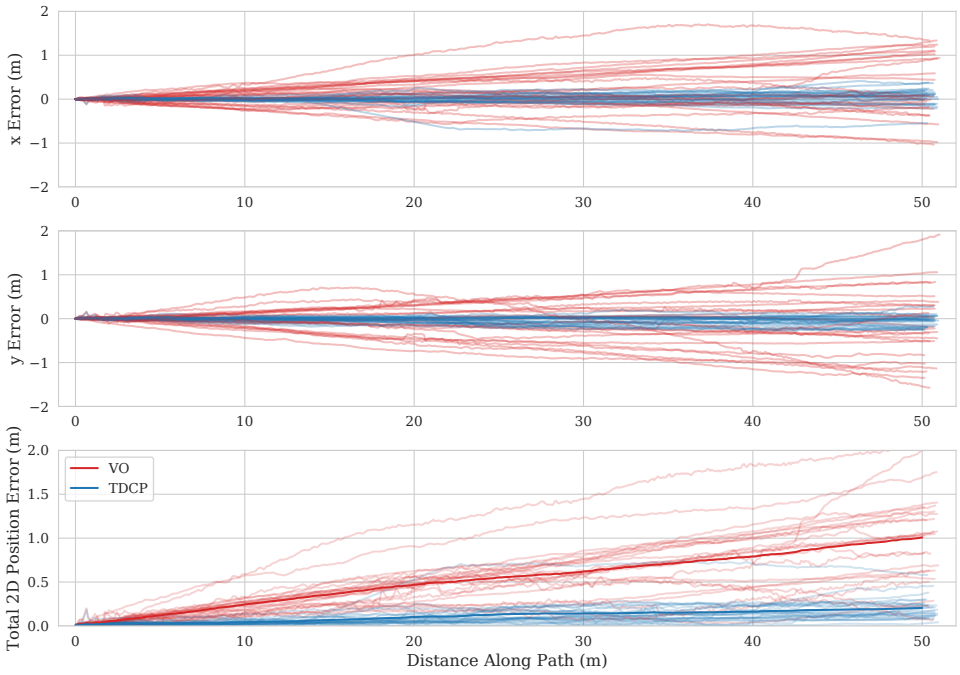

Figure 12. Comparison of VO and TDCP-based single-receiver GPS odometry position drift. The fainter lines represent individual trajectories while the darker line plots the average error for the algorithm.

Table 4. Comparison of using GPS vs VO as localization prior in VT&R under nominal conditions (experiment 1).

	Mean Lateral Localization Error (m)	
Path	**VO Prior**	**GPS Prior**
1	**0.022**	**0.022**
2	0.081	**0.020**
3	0.036	**0.035**
4	0.025	**0.024**
5	0.029	**0.028**
6	**0.028**	**0.028**
7	**0.026**	**0.026**

5.3. Performance in VT&R

5.3.1. Experiment 1: Nominal Conditions

Figure 13 shows the results for experiment 1 on one shorter path (path 3) and one longer path (path 7). Path 3 had the most challenging conditions and shows the worst-case performance of our method. The error plotted in this section is the difference between our estimated path-tracking offset from localization compared to the ground truth path-tracking error. We show the lateral error in the vehicle frame as this is most important for maintaining accurate path tracking. Table 4 summarizes the results for this experiment across all paths.

In experiment 1, the full stereo stream is available to both algorithms and no adverse conditions are simulated. Both the VO and the GPS odometry methods provided similar performance. Three of the paths had both methods giving the same mean localization error while the GPS odometry performed better on the other four. However, the difference was very small. This was expected as

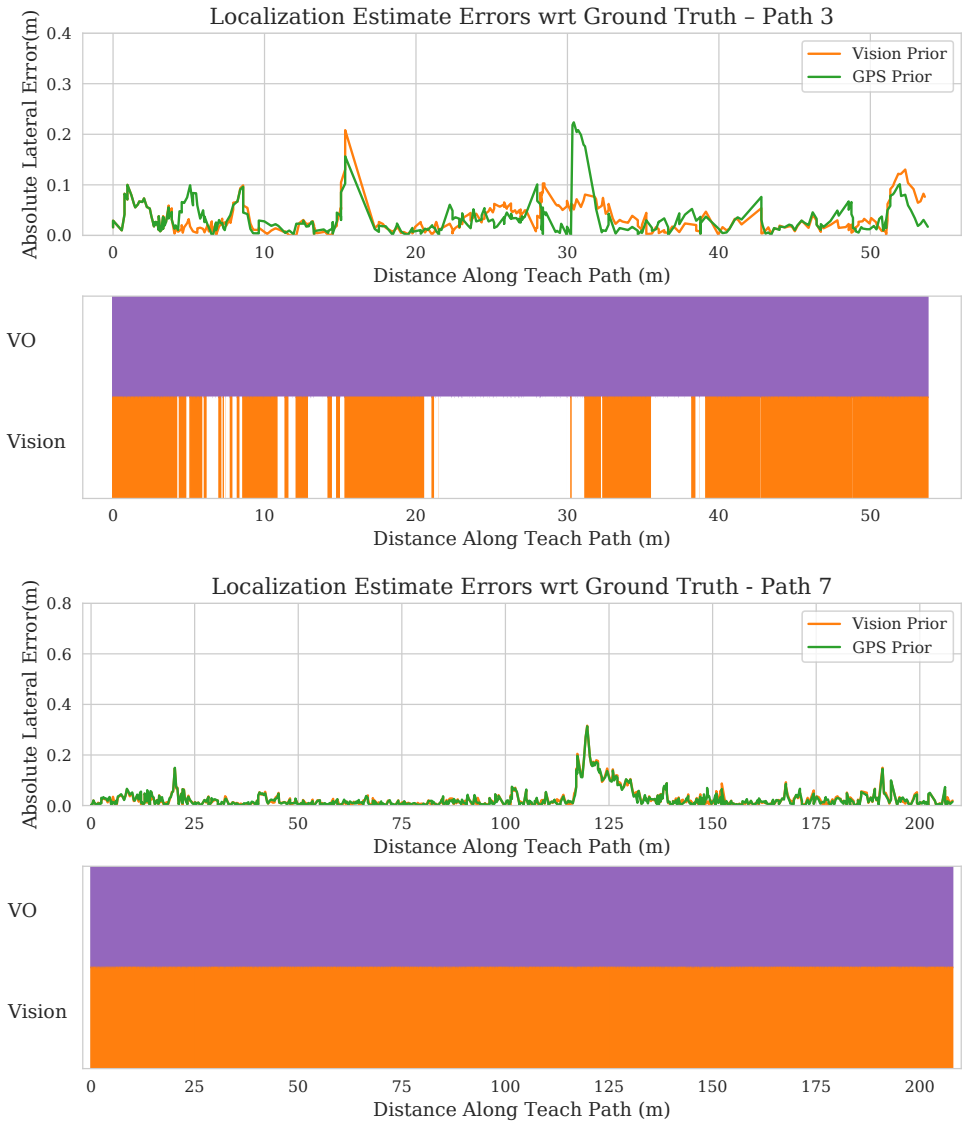

Figure 13. The results of experiment 1 on one of the short paths (top) and one of the long paths (bottom). The upper subplot shows the errors along the path for each algorithm while the lower subplot shows where visual odometry (purple) and visual localization (orange) were successful. In this experiment, full use of vision was available, though some naturally occurring localization failures were seen on path 3 due to the nighttime conditions.

the relative weight of the prior is reduced when many visual features are available for localization. Therefore, under conditions where VT&R is already highly successful, our addition does not degrade that performance. For path 3, the manual driving and nighttime conditions created some natural localization failures, as seen in the bottom subplot of Figure 13. However, both methods were able to bound errors to a reasonable level for path-tracking.

5.3.2. Experiment 2: Dead Reckoning

VT&R attempts visual localization during a repeat run when the number of feature matches with respect to the map reaches a minimum threshold. In experiment 2, we simulate a prolonged section where visual localization is not possible by forcing the number of feature matches to be less than

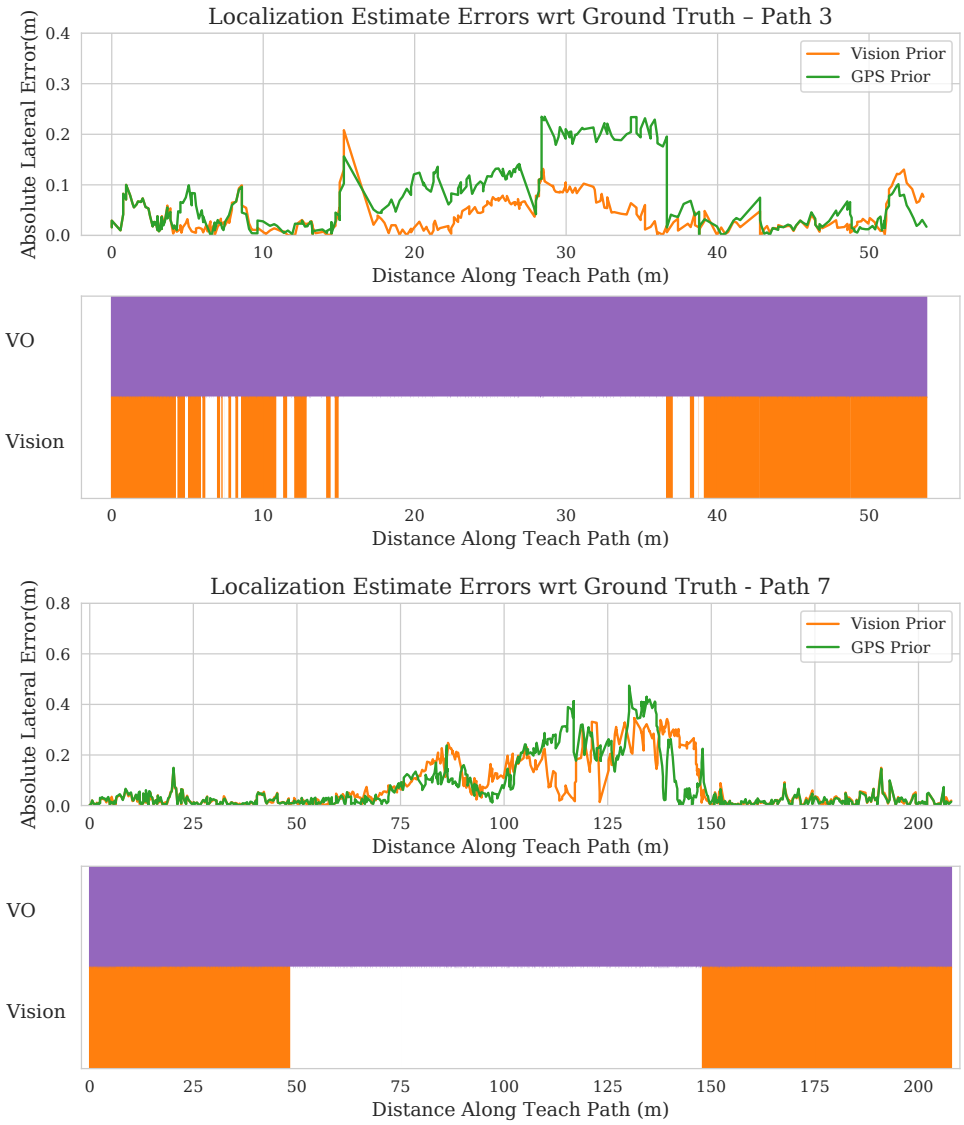

Figure 14. The results of experiment 2 on one of the short paths (top) and one of the long paths (bottom). The upper subplot shows the errors along the path for each algorithm while the lower subplot shows where visual odometry (purple) and visual localization (orange) were successful. In this experiment, we prevented the robot from using visual localization in the middle section of the repeat runs, forcing it to rely on its prior.

the threshold. Therefore the algorithm must instead rely heavily on its prior. The feature matches used to perform VO are not affected. The visual localization dropout lasts for approximately 20 m of driving on the shorter paths and 100m on the longer ones. These results can be seen in Figure 14 and are summarized in Table 5.

We find the localization error grows larger in the section in which we prevent visual localization, as expected. This is true for both methods and on all paths. For four of the seven paths, the mean error with the GPS prior is higher than the VO prior but, they are of similar magnitudes for all paths. Both algorithms do well enough that when the robot is allowed to localize again, it still knows where it is along the path and the localization error quickly drops. None of the runs would be considered a failure as the robot completed the path in all cases. Therefore, when VO is successful, either method is suitable to use as a prior when visual localization struggles.

Table 5. Comparison of using GPS vs. VO as localization prior in VT&R when visual localization is partially unavailable (experiment 2).

	Mean Lateral Localization Error (m)	
Path	**VO Prior**	**GPS Prior**
1	**0.031**	0.054
2	0.104	**0.030**
3	**0.042**	0.079
4	**0.046**	0.134
5	0.217	**0.091**
6	**0.171**	0.195
7	0.082	**0.080**

Table 6. Comparison of using GPS vs VO as localization prior in VT&R when visual odometry is partially unavailable (experiment 3).

	Mean Lateral Localization Error (m)	
Path	**VO Prior**	**GPS Prior**
1	6.767	**0.077**
2	3.704	**0.036**
3	1.525	**0.092**
4	10.646	**0.375**
5	10.283	**0.248**
6	10.932	**0.251**
7	11.753	**0.152**

5.3.3. Experiment 3: Camera Dropout

In experiment 3, we again simulate a sensor dropout over the middle section of each repeat, but this time we replace the left camera images with black images instead of only blocking visual localization. As a result, stereo VO also fails. Figure 15 and Table 6 show our results.

In this experiment, the vision-only pipeline is less successful. Without GPS, the robot can rely on its motion model for a short period but quickly becomes lost without the proper means for dead-reckoning. Even when VO recovers, the robot is unsure of where it is in relation to the path and is not able to relocalize. This produces the large mean errors in Table 6. Every run would be considered a failure as the robot ends up completely off the path. When using the GPS prior, we do not rely on VO while GPS odometry is available. As a result, errors are similar in magnitude to experiment 2. With GPS, the robot quickly recovers and accurately localizes when the camera images are restored. Our addition of GPS odometry to the VT&R pipeline prevented failure on every run.

6. Conclusion and Future Work

In this paper, we described a method for highly accurate odometry using a single GPS receiver. While GPS users are typically concerned with the absolute accuracy of their positioning algorithms, we recognize that only relative accuracy is required for odometry. Relaxing this constraint meant we could cancel many of the temporally correlated error sources affecting GPS and get better displacement estimates. We compared the performance of our single-receiver GPS odometry with stereo VO on the same set of test trajectories. The novel contributions of this work are as follows:

(a) we detailed a practical TDCP-based odometry algorithm complete with a motion model for use on a UGV,
(b) we provided the first direct comparison of single-receiver GPS and VO estimation,

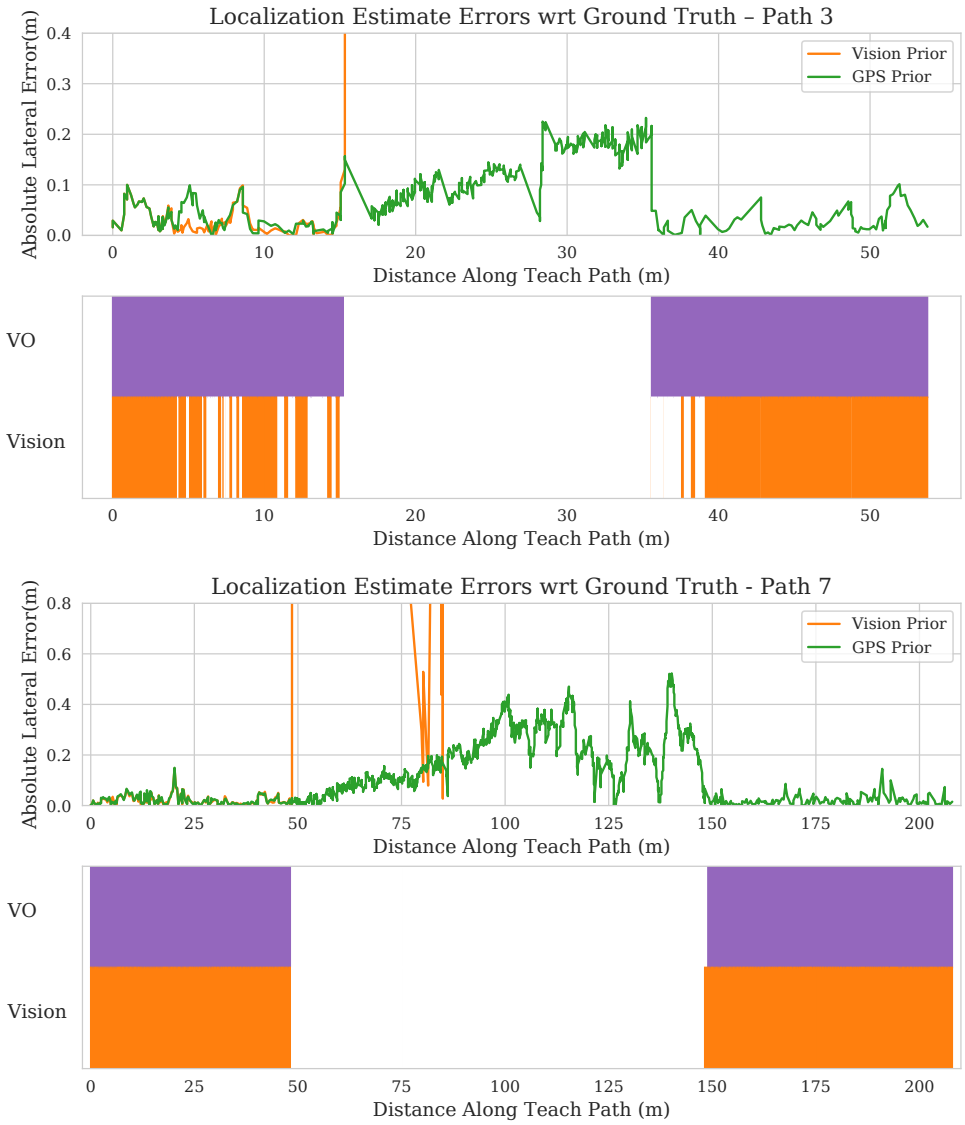

Figure 15. The results of experiment 3 on one of the short paths (top) and one of the long paths (bottom). The upper subplot shows the errors along the path for each algorithm while the lower subplot shows where visual odometry (purple) and visual localization (orange) were successful. The gap in the purple bar shows where the camera failure was simulated.

(c) we showed how our TDCP-odometry can be used to complement VT&R's localization pipeline leading to an improved autonomy rate.

We believe TDCP odometry is an effective navigation technique and is underutilized in robotics compared to other odometry methods. To show this, we simultaneously collected a large set of GPS data and stereo imagery from a ground robot driving outdoors. We evaluated our TDCP-based single-receiver, single-frequency GPS odometry algorithm against a proven stereo VO pipeline in the first known experiment of this kind. The results showed the GPS odometry produced far smaller positional errors with respect to the RTK ground truth. TDCP odometry is a good alternative to VO for outdoor navigation. VO is still preferred in areas where occlusions or other sources of GPS signal interference are a frequent issue.

For added robustness, the two sensors may be combined. In Section 3.5, we showed how one can use GPS as a backup or alternative for VO in VTR3. GPS odometry provides an accurate prior allowing localization to achieve similar performance as with VO in good conditions and preventing total localization failures in challenging scenarios. As this is run in a separate node, the details are abstracted from the main stereo camera pipeline allowing our changes to easily be used in future varieties of VT&R such as those using radar or lidar as the primary sensor. A single GPS receiver is the only additional equipment required to obtain this improved robustness of our system when vision struggles. Beyond VT&R, CPO can be run as a standalone GPS odometry package and is available open-source at `https://github.com/utiasASRL/cpo`. To the best of our knowledge, this is the only widely available TDCP software package and one of the few open-source navigation projects developed in ROS2. The project provides accurate odometry with final drift rates averaging just 0.38% in our experiments.

One interesting observation from the results of Section 5.1 was that the position errors with respect to ground truth vary more linearly than we would expect in a random walk, as odometry is often modelled. We hypothesize this is due to the uncorrected error sources in the carrier phase double-difference equation [Equation (3)] varying smoothly. We might expect these error sources to be autocorrelated given they are a function of satellites moving at near-constant velocity sending signals through an atmosphere that changes gradually. An interesting extension might be to use other sensors or a motion model to estimate and correct for this linear bias.

Though we did not explore incorporating additional GNSS constellations in our algorithm, it is likely their use could improve positioning accuracy even further by increasing the number of satellites available. Improved odometry allows robots to build better maps and safely drive further when localization against a map is challenging. This would be beneficial for many robots and robotic systems.

ORCID

Benjamin Congram https://orcid.org/0000-0002-9116-5294
Timothy D. Barfoot https://orcid.org/0000-0003-3899-631X

References

Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W. (2013). Robust map optimization using dynamic covariance scaling. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 62–69. IEEE.

Anderson, S. and Barfoot, T. D. (2015). Full steam ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on se(3).

Balard, N., Santerre, R., Cocard, M., and Bourgon, S. (2006). Single gps receiver time-relative positioning with loop misclosure corrections. *GPS Solutions*, 10(1):56–62.

Barfoot, T. D., Tong, C. H., and Särkkä, S. (2014). Batch continuous-time trajectory estimation as exactly sparse gaussian process regression. In *Robotics: Science and Systems*, volume 10. Citeseer.

Cao, S., Lu, X., and Shen, S. (2022). Gvins: Tightly coupled gnss–visual–inertial fusion for smooth and consistent state estimation. *IEEE Transactions on Robotics*.

Chong, K. S. and Kleeman, L. (1999). Feature-based mapping in real, large scale environments using an ultrasonic array. *Int. J. Robotics Res.*, 18(1):3–19.

Churchill, W. and Newman, P. (2013). Experience-based navigation for long-term localisation. *Int. J. Robotics Res.*, 32(14):1645–1661.

Collins, P., Langley, R., and LaMance, J. (1996). Limiting factors in tropospheric propagation delay error modelling for gps airborne navigation. *Proc. Inst. Navig. 52nd Ann. Meet*, 3.

Congram, B. and Barfoot, T. D. (2021). Relatively lazy: Indoor-outdoor navigation using vision and gnss. In *2021 18th Conference on Robots and Vision (CRV)*, pages 25–32.

Cooper, S. and Durrant-Whyte, H. (1994). A kalman filter model for gps navigation of land vehicles. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 1, pages 157–163 vol.1.

Dai, Z., Knedlik, S., and Loffeld, O. (2009). Instantaneous triple-frequency gps cycle-slip detection and repair. *International Journal of Navigation & Observation*.

Ding, W. and Wang, J. (2011). Precise velocity estimation with a stand-alone gps receiver. *The Journal of Navigation*, 64(2):311–325.

Furgale, P. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560.

Imperoli, M., Potena, C., Nardi, D., Grisetti, G., and Pretto, A. (2018). An effective multi-cue positioning system for agricultural robotics. *IEEE Robotics and Automation Letters*, 3(4):3685–3692.

Kaplan, E. and Hegarty, C. (2005). *Understanding GPS: principles and applications*. Artech house.

Kim, D. and Langley, R. B. (2002). Instantaneous real-time cycle-slip correction for quality control of gps carrier-phase measurements. *Navigation*, 49(4):205–222.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces.

Klobuchar, J. A. (1987). Ionospheric time-delay algorithm for single-frequency gps users. *IEEE Transactions on aerospace and electronic systems*, (3):325–331.

Liu, Z., Ji, S., Chen, W., and Ding, X. (2013). New fast precise kinematic surveying method using a single dual-frequency gps receiver. *Journal of Surveying Engineering*, 139(1):19–33.

Marshall, J. A., Barfoot, T. D., and Larsson, J. (2008). Autonomous underground tramming for center-articulated vehicles. *J. Field Robotics*, 25(6-7):400–421.

Michaud, S. (2001). *Investigation d'une nouvelle approche de positionnement relatif temporel avec GPS et avec GLONASS*.

Michaud, S. and Santerre, R. (2001). Time-relative positioning with a single civil gps receiver. *GPS Solutions*, 5(2):71–77.

Niell, A. (1996). Global mapping functions for the atmosphere delay at radio wavelengths. *Journal of Geophysical Research: Solid Earth*, 101(B2):3227–3246.

Ohno, K., Tsubouchi, T., Shigematsu, B., and Yuta, S. (2004). Differential gps and odometry-based outdoor navigation of a mobile robot. *Advanced Robotics*, 18(6):611–635.

Paton, M., MacTavish, K., Warren, M., and Barfoot, T. D. (2016). Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1918–1925.

Pierce, D., Martin, S., and Bevly, D. M. (2017). Opportunistic landmark registration for long distance relative path following. In *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, pages 2560–2573.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

Scherzinger, B. M. (2006). Precise robust positioning with inertially aided rtk. *Navigation*, 53(2):73–83.

Seeber, G. (2008). *Satellite geodesy: foundations, methods, and applications*. Walter de gruyter.

Sibley, G., Mei, C., Reid, I. D., and Newman, P. (2009). Adaptive relative bundle adjustment. In Trinkle, J., Matsuoka, Y., and Castellanos, J. A., editors, *Robotics: Science and Systems V, University of Washington, Seattle, USA, June 28 - July 1, 2009*. The MIT Press.

Soon, B. K., Scheding, S., Lee, H.-K., Lee, H.-K., and Durrant-Whyte, H. (2008). An approach to aid ins using time-differenced gps carrier phase (tdcp) measurements. *Gps Solutions*, 12(4):261–271.

Subirana, J. S., Zornoza, J. J., and Hernández-Pajares, M. (2013). Gnss data processing. volume 1: Fundamentals and algorithms. *ESA Communications ESTEC, PO Box*, 299:2200.

Sukkarieh, S., Nebot, E., and Durrant-Whyte, H. (1999). A high integrity imu/gps navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15(3): 572–578.

Suzuki, T. (2020). Time-relative RTK-GNSS: GNSS loop closure in pose graph optimization. *IEEE Robotics Autom. Lett.*, 5(3):4735–4742.

Traugott, J. (2011). *Precise flight trajectory reconstruction based on time-differential GNSS carrier phase processing*. PhD thesis, Technische Universität München.

Travis, W. (2010). *Path Duplication Using GPS Carrier Based Relative Position for Automated Ground Vehicle Convoys*. PhD thesis.

Ulmer, K., Hwang, P., Disselkoen, B., and Wagner, M. (1995). Accurate azimuth from a single plgr+ gls dod gps receiver using time relative positioning. In *Proceedings of the 8th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1995)*, pages 1733–1741.

van Graas, F. and Lee, S.-W. (1995). High-accuracy differential positioning for satellite-based systems without using code-phase measurments. *NAVIGATION, Journal of the Institute of Navigation*, 42(4):605–618.

Wen, W., Bai, X., Kan, Y. C., and Hsu, L.-T. (2019). Tightly coupled gnss/ins integration via factor graph and aided by fish-eye camera. *IEEE Transactions on Vehicular Technology*, 68(11):10651–10662.

Wen, W. and Hsu, L.-T. (2021). Towards robust gnss positioning and real-time kinematic using factor graph optimization. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5884–5890. IEEE.

Wendel, J., Meister, O., Monikes, R., and Trommer, G. (2006). Time-differenced carrier phase measurements for tightly coupled gps/ins integration. In *Proceedings of IEEE/ION PLANS 2006*, pages 54–60.

Wendel, J., Obert, T., and Trommer, G. F. (2003). Enhancement of a tightly coupled gps/ins system for high precision attitude determination of land vehicles. In *Proceedings of the 59th Annual Meeting of The Institute of Navigation and CIGTF 22nd Guidance Test Symposium (2003)*, pages 200–208.

Wendel, J. and Trommer, G. F. (2004). Tightly coupled gps/ins integration for missile applications. *Aerospace Science and Technology*, 8(7):627–634.

Williams, S. B. (2001). *Efficient Solutions to AutonomousMapping and Navigation Problems*. PhD thesis, The University of Sydney.

Yu, Y., Gao, W., Liu, C., Shen, S., and Liu, M. (2019). A gps-aided omnidirectional visual-inertial state estimator in ubiquitous environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7750–7755. IEEE.

Zhao, S., Chen, Y., Zhang, H., and Farrell, J. A. (2014). Differential gps aided inertial navigation: A contemplative realtime approach. *IFAC Proceedings Volumes*, 47(3):8959–8964.

Zhao, Y. (2016). Applying time-differenced carrier phase in nondifferential gps/imu tightly coupled navigation systems to improve the positioning performance. *IEEE Transactions on Vehicular Technology*, 66(2):992–1003.

How to cite this article: Congram, B., & Barfoot, T. D. (2022). Field testing and evaluation of single-receiver GPS odometry for use in robotic navigation. *Field Robotics, 2*, 1849–1873.