

Global Pose Estimation with Limited GPS and Long Range Visual Odometry

Joern Rehder, Kamal Gupta, Stephen Nuske, and Sanjiv Singh

Abstract—Here we present an approach to estimate the global pose of a vehicle in the face of two distinct problems; first, when using stereo visual odometry for relative motion estimation, a lack of features at close range causes a bias in the motion estimate. The other challenge is localizing in the global coordinate frame using very infrequent GPS measurements. Solving these problems we demonstrate a method to estimate and correct for the bias in visual odometry and a sensor fusion algorithm capable of exploiting sparse global measurements. Our graph-based state estimation framework is capable of inferring global orientation using a unified representation of local and global measurements and recovers from inaccurate initial estimates of the state, as intermittently available GPS information may delay the observability of the entire state. We also demonstrate a reduction of the complexity of the problem to achieve real-time throughput. In our experiments, we show in an outdoor dataset with distant features where our bias corrected visual odometry solution makes a five-fold improvement in the accuracy of the estimated translation compared to a standard approach. For a traverse of 2km we demonstrate the capabilities of our graph-based state estimation approach to successfully infer global orientation with as few as 6 GPS measurements and with two-fold improvement in mean position error using the corrected visual odometry.

I. INTRODUCTION

In this paper we study the problem of localizing a vehicle in a global coordinate frame given two distinct challenges found in outdoor environments. One challenge is the features detected for a stereo visual odometry system can be at a significant distance from the camera and introduce a bias in the motion estimate. The other challenge we address is maintaining a coherent and accurate global solution with intermittent GPS coverage. We focus our experimental results on a mapping application in riverine environments where both problems can be found, but these issues are also pertinent to other environments and applications.

We extend our prior results in river mapping with an autonomous rotorcraft [1] and present new approaches to improve accuracy and efficiency. Our application has tight constraints on power consumption and weight, which have significant implications on the sensor payload. We therefore heavily rely on a light weight stereo camera for visual odometry and fuse its motion estimates with readings from MEMS inertial sensors and position data recorded with a consumer grade GPS receiver. In earlier work [1], we

employed a common approach to stereo visual odometry with demonstrated accuracy in automotive applications [2]. We found that the algorithm performed well in preceding experiments in urban environments, but it exhibited a significant bias for sequences recorded on a river. In this work, we will show that riverine environments are characterized by a distinctly different distribution of visual features and that this lack of structure at close range results in a consistent underestimation of the translation for the class of visual odometry algorithms used in our experiments. We will introduce an approach to estimating and correcting this bias and demonstrate improvements of an approximate five-fold reduction in the bias.

Besides suffering from impaired visual odometry, autonomous aerial vehicles in riverine environment will also have to cope with intermittent GPS availability caused by tall vegetation along the banks that blocks satellite reception. Whenever available, global position estimates provide valuable information to reduce drift in the pose estimation and constraint the vehicle pose in a global coordinate frame.

We therefore define the following set of requirements for our full 6 degrees of freedom (DoF) pose estimation

- consistently fuse the inputs of stereo visual odometry, inertial measurements, and sparse GPS information into a single pose estimate with real-time throughput
- have the ability to deal with significant changes of the pose in the global coordinate frame defined by GPS and avoid discontinuities in the vehicle path

These requirements have substantial implications on the pose estimation. Having to avoid discontinuities in the path, the pose estimation has to be able to adjust a history of poses. The length of this history is determined by GPS availability, as the pose estimation must include multiple measurements to infer global orientation from GPS information. On the other hand, demanding real-time throughput requires a constant reduction of the problem to keep it computational feasible.

In this paper, we describe a pose estimation method that treats the problem as a graph-based non linear optimization. It employs a unified representation of global and local constraints and continuously reduce the complexity of the problem.

We demonstrate the performance of the algorithm on a challenging dataset captured in a riverine environment, showing that by applying our bias correction to stereo visual odometry and fusing it with inertial measurements and as little as 6 GPS locations over a distance of about 2 km, we can reconstruct a path that closely replicates the ground truth

J. Rehder was with the TUHH, Germany. He is now with the ETH Zurich, Switzerland joern.rehder@mavt.ethz.ch

K. Gupta is with the Indian Institute of Technology Delhi, India, kamalgupta308@gmail.com

S. Nuske and S. Singh are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A., {nuske,ssingh}@cmu.edu

recorded with a high accuracy GPS. Even by incorporating so few GPS measurements into the optimization, we can still estimate the global position and orientation of the path with a mean error of 5m from truth. To our knowledge, no other state of the art approach fuses GPS measurements at such low rates, while estimating consistent paths with real-time performance.

We first discuss related work in Sec. II, we present a method to estimate and correct the bias in visual odometry in Sec. III, describe our graph-based global state estimation framework in Sec. IV and provide results and conclusions of our work in Sec. V and Sec. VI.

II. RELATED WORK

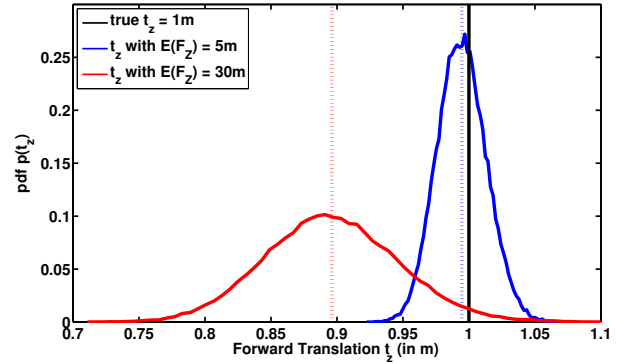
Here we focus on the related work that has been done both on stereo visual odometry and in the field of state estimation from multiple sensor sources.

Visual odometry is a well established tool to estimate the motion for vehicles both travelling on the ground [3], [4], [5], [6], [7], [2] and aerial vehicles travelling near to ground [8]. Most experimental results presented along with the aforementioned approaches were obtained in environments with visual features in a short range. Distant features introduce a bias in stereo triangulation [9], [10]. The bias is also apparent in stereo visual odometry solutions and there have been recent attempts to correct for the bias after executing a trajectory with the use of GPS readings [7]. In this work, we address the implications of this bias on visual odometry for long range features and we present an approach to estimate and correct the bias for each frame using visual information alone.

State estimation with a focus on mapping is a well studied problem in robotics. In the past, recursive filters have been successfully demonstrated for a variety of applications (e.g. [11], [12]). Recent research [13] suggests that recursive filtering performs inferiorly compared to optimization approaches for many applications. Furthermore, we expect the fusion of sparsely available GPS information to result in large changes in the state estimation, which makes repeated relinearizations necessary. Our approach is similar to FrameSLAM [14] in that it fuses inertial measurements with visual odometry, but it consistently considers GPS readings in the state estimation as well. A similar approach [15] integrates GPS readings into the position estimation using an EKF, though it remains unclear whether the approach can deal with sparse and seriously impaired GPS reception. GraphSLAM [16] is capable of fusing GPS measurements into the SLAM problem, but it constitutes an offline solution to the problem not intended to perform in real-time. Similarly, our general approach marks a sub-class of problems that can be addressed with the g^2o framework [17], but it additionally incorporates a graph reduction scheme comparable to [18] to keep the problem computationally tractable. Furthermore, this work makes use of the sparse Levenberg-Marquardt framework presented in [19], which provides an interface to a similar set of efficient sparse solvers as the g^2o framework.



(a) Feature distribution



(b) Bias in long range visual odometry

Fig. 1. (a) A typical riverine environment overlaid with the spatial distribution of matches and their temporal flow. The features are color coded by their depth. Note the absence of any salient, static, visual features in close proximity of the vehicle. (b) Simulation results for stereo visual odometry with feature distributions resembling an urban and a riverine environment. The black line indicates true motion, the blue solid line marks the distribution of estimates from 50,000 random trials with a mean feature distance of 5m, while the red solid line shows the distribution of estimates for a mean feature distance of 30m. Dotted lines indicate the mean of the distributions. The graph illustrates that features at long range create a bias in visual odometry towards underestimating the motion. We fix the simulation to match our experimental setup. Note that improved resolution, baseline and pixel noise, would improve the underestimation problem. However, no matter the parameters, there will be a feature configuration that creates a biased estimate.

III. LONG RANGE VISUAL ODOMETRY

We employ a stereo visual odometry algorithm to give accurate and consistent local motion estimates and provide a solution to errors introduced by operating a visual odometry system at a long range from the nearby environment.

Recent visual odometry systems [2], [4] report high accuracy with errors less than 2% of distance travelled and sometimes as low as a fraction of a percent [6]. However, these high accuracy results all have one commonality which is that the evaluation has been conducted in environments where features are mostly at close range to the camera.

Fig. 1(a) shows the distribution of feature matches in a riverine environment, where features are predominantly from the shoreline and often at a significant range from the camera. We find in our tests that the conventional approaches for

visual odometry do not only have low accuracy when the features are at a long range from the camera, but also exhibit a significant bias towards underestimating the motion. This bias is depicted in Fig. 1(b), where visual odometry was simulated for an experiment with features at close range versus one with features at long range.

To understand the issue of stereo visual odometry at long ranges, we study the core component common to all approaches which is triangulating the set of 2D image feature locations in the left image $F_{[u,v]}^l$ and right image $F_{[u,v]}^r$ of the stereo pair, into 3D coordinates $F_{[x,y,z]}^i$:

$$d = F_u^l - F_u^r \quad (1)$$

$$F_z = \frac{bf}{d} \quad (2)$$

$$F_x = F_z(F_u^l - c_u)/f \quad (3)$$

$$F_y = F_z(F_v^l - c_v)/f \quad (4)$$

where F_u and F_v is the horizontal and vertical image coordinate of the feature, f is the focal length of the camera in pixels and c_u, c_v is the center of projection. In the past the error in triangulation was often modelled with a zero-mean Gaussian distribution [20] but more recently it has been established in the literature [9], [10] that the error is a non-Gaussian distribution with a heavy tail, that introduces a bias in the triangulation. The non-Gaussian distribution can be verified in simulation with a Gaussian noise model, γ , on the 2D image feature locations $F^{[u,v]}$. The bias in triangulation becomes more pronounced with range of the features from the camera, specifically this is when the noise in the extracted feature locations, γ , approaches the signal, the feature disparity, d , from Eq. 1.

We find the bias in triangulation when placed in context of visual odometry creates an underestimation of the motion of camera. The issue has been discovered previously in [7] and the authors present a method to estimate and correct the bias after-the-fact using GPS readings. However, when moving through an environment the feature configuration changes dramatically and, hence the error changes rapidly which means a single correction factor for the bias computed after travelling a trajectory is not sufficient.

In this work we demonstrate how to estimate the bias in visual odometry for each frame at the time of computing and ideally without relying on GPS or other positioning sources. The approach first takes the motion solution from a generic stereo visual odometry algorithm. We believe any visual odometry algorithm will suffer from the underestimation issue, for our specific implementation we adopt the approach of [2] that uses commonly deployed iterative minimization on reprojection error to solve for the motion. We denote the visual odometry algorithm, $g(\cdot)$, as a function of the features extracted in the two stereo pairs of the current frame, i , and previous frame, $i-1$:

$$[R_o, t_o] = g(F^{i-1}, F^i) \quad (5)$$

the resulting rotation, R_o , and translation, t_o , solution has an

unknown bias, k , from the true motion $[R_w, t_w]$ as follows:

$$[R_w, t_w] = [R_o, kt_o] \quad (6)$$

To estimate the bias we generate a new solution $[\bar{R}, \bar{t}]$ by simulating a stereo camera at the pose of the initial solution, $[R_o, t_o]$, and projecting the triangulated 3D feature locations from the previous stereo-pair, $F_{[x,y,z]}^{i-1}$, onto the simulated camera plane, as 2D coordinates $\bar{F}_{[u,v]}^i$:

$$F_{[x,y,z]}^i = \left(R_o^T F_{[x,y,z]}^{i-1} - R_o^T t_o \right) \quad (7)$$

$$\bar{F}_u^i = \gamma + (f F_x^i / F_z^i + c_u)$$

$$\bar{F}_v^i = \gamma + (f F_y^i / F_z^i + c_v) \quad (8)$$

where R_o^T is the transpose of the rotation matrix and γ is the pixel noise which we inject as a random sample from a Gaussian distribution which we fix at 0.5 pixel standard deviation.

Using the features simulated on our second stereo pair, \bar{F}^i , and the features originally extracted in the first stereo pair, F^{i-1} , we compute a new motion estimate, $[\bar{R}, \bar{t}]$, with:

$$[\bar{R}, \bar{t}] = g(F^{i-1}, \bar{F}^i) \quad (9)$$

To derive a stable solution, we repeat the simulation several times, by generating new random samples of γ and generate a set of J motion solutions, from which we extract the mean:

$$[\bar{R}, \bar{t}] = \sum_{j=1}^J \frac{g(F^{i-1}, \bar{F}_j^i)}{J} \quad (10)$$

We then compute a bias estimate \bar{k} as the error between the norm of the original and simulated translation estimate:

$$\bar{k} = \frac{\|t_o\|}{\|\bar{t}\|} \quad (11)$$

We apply the bias estimate to derive our corrected motion estimate:

$$[R_c, t_c] = [R_o, \bar{k}t_o] \quad (12)$$

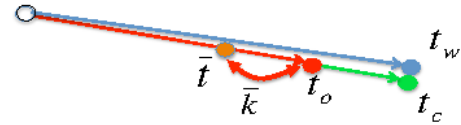


Fig. 2. Diagram of the bias correction. The actual motion t_w is underestimated by the visual odometry algorithm (t_o). We simulate a camera at t_o using the same feature distribution as estimated in the stereo pair and generate a new estimate \bar{t} . We compute the bias \bar{k} between original and simulated estimates, depicted by the curved red arrows. We assume the bias at the initial solution is a good approximation of the actual bias and generate a corrected motion estimate t_c , represented by the green arrow.

Our corrected motion estimate appears to hold true subject to two assumptions. First is that our pixel noise model γ is accurate. Secondly, the bias in motion, k , can be approximated by the bias at our original estimate $[R_o, t_o]$, that is to say the bias is locally smooth in the solution space. The process is summarized in Fig. 2, where the red curved arrow

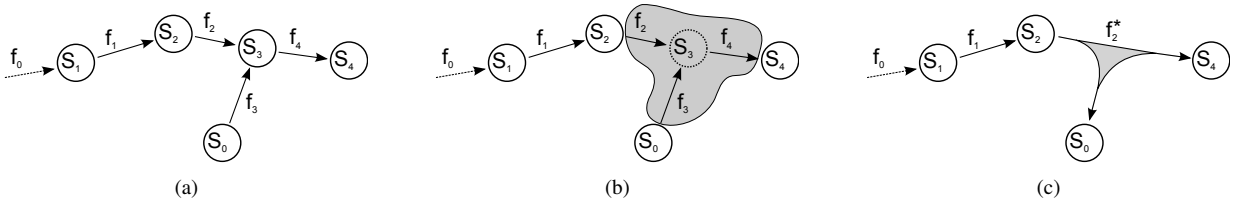


Fig. 3. This sequence of figures depicts the procedure of reducing the dimension of a pose graph by the node s_3 . Fig. (a) shows a subsection of the pose graph, where the nodes s_M represent the vehicle pose at different time steps t_k , while s_0 marks a virtual zero pose. The poses are constrained by measurements, modelled as edges in the graph. In Fig. (b), the sub-graph consisting of node s_3 and all directly attached constraints is removed from the graph. It is replaced by a single constraint resembling the same dependence, which depends on all nodes that have previously been connected the marginalized node (see Fig. (c)). Section IV-B describes the operations involved in the marginalization of nodes in more detail.

represents the computed bias and the green arrow represents the bias being applied to generate the corrected estimate.

Later in the results section we compare our corrected estimate against the original output of a standard visual odometry system [2].

IV. OPTIMAL POSE ESTIMATION WITH SPARSE GLOBAL POSITION INFORMATION

Our approach to global pose estimation is closest to [18] and [14] in the sense that it treats the problem as a non linear optimization over a history of vehicle poses with a graphical representation. The graph consists of a set of nodes that represent past vehicle poses at discrete points in time. Sensor measurements induce constraints on these vehicle poses, which are modelled as edges that connect the nodes of the graph. In general, the inputs to our pose estimation can be classified into local measurements, that induce edges between two successive nodes (e.g. stereo visual odometry, relative orientation from gyroscopes), and global measurements, which impose constraints on the pose in a global coordinate frame (e.g. GPS readings, inclination from accelerometers). By introducing a virtual zero node at the origin of the coordinate system into the graph, both classes of constraints can be treated in a unified way [21].

We make the assumption that we can model sensor measurements x_k as a function h_k of the pose at exactly two points in time $s_1(k)$ and $s_2(k)$ or the pose $s_2(k)$ and the fictitious zero pose s_0 . Furthermore, we assume that measurement noise can be modelled as a zero-mean Gaussian distributed random variable v .

$$x_k = h_k(s_1(k), s_2(k)) + v_k \quad (13)$$

By minimizing the cost function

$$\sum_{k=1}^n (h_k(s_1(k), s_2(k)) - x_k)^T C^{-1} (h_k(s_1(k), s_2(k)) - x_k) \quad (14)$$

a set of vehicle poses can be determined, which is optimal with respect to the previously stated assumptions. With $C^{-1} = Q^T Q$, the optimization can be treated as a non linear least square problem

$$\sum_{k=1}^n \|Q(h_k(s_1(k), s_2(k)) - x_k)\|^2 \quad (15)$$

The state S of the overall system is formed by concatenating the individual pose information s_i , collected from the nodes

in the graph. Each constraint contributes $f_k(s_1(k), s_2(k)) = Q(h_k(s_1(k), s_2(k)) - x_k)$ to the overall set of non linear equations $f(S)$. We employ a sparse Levenberg-Marquardt library [19] to efficiently solve for the optimization problem.

A. Pose Parametrization and Sensor Models

The pose is parametrized as

$$s = [\Psi, t]^T = [\phi, \theta, \psi, t_x, t_y, t_z]^T \quad (16)$$

where $\Psi = [\phi, \theta, \psi]^T$ denotes Euler angles parametrizing the orientation, and $t = [t_x, t_y, t_z]^T$ parametrizes the position in a global coordinate frame. Note that there exist different conventions for Euler angles, which expose different singularities. We chose Euler angles for which the singularities coincide with kinematically infeasible orientations of the rotorcraft in normal operation.

Our current implementation uses the stereo visual odometry presented in [2] and corrects for the bias according to Sec. III to provide the 6 DoF motion that the camera underwent in between taking two sets of stereo images along with the uncertainty of the estimate. Nodes are added to the graph when visual odometry is available, using its 6 DoF transformation to obtain an initial estimate of the current vehicle pose, while the optimization step is run only after a few nodes have been added to the graph. With $R(\Psi)$ denoting the $\mathbb{R}^3 \mapsto \mathbb{R}^{3 \times 3}$ mapping of Euler angles Ψ to a rotation matrix, and $R^{-1}(A)$ denoting the inverse mapping from a rotation matrix A to Euler angles, visual odometry constraints are defined as

$$f_{vo}(s_i, s_j) = Q_{vo} \left(T(s_i, s_j) - \begin{bmatrix} \Psi_{vo} \\ t_{vo} \end{bmatrix} \right) \quad (17)$$

$$= Q_{vo} \begin{bmatrix} R^{-1}(R(\Psi_i)^T R(\Psi_j)) - \Psi_{vo} \\ R(\Psi_i)^T (t_j - t_i) - t_{vo} \end{bmatrix} \quad (18)$$

where $x = [\Psi_{vo}, t_{vo}]^T$ denotes a transformation, measured by visual odometry in the coordinate frame of s_i . The function $T(s_i, s_j)$ transforms pose s_j into the coordinate system of pose s_i .

Gyroscope constraints are defined respectively on the orientations of successive poses:

$$f_{gyro}(s_i, s_j) = Q_{gyro} [R^{-1}(R(\Psi_i)^T R(\Psi_j)) - \Psi_{gyro}] \quad (19)$$

where Ψ_{gyro} is obtained from numerical integration of angular velocities as presented in [22]. For simplicity, we assume

that the coordinate frames of the stereo camera and inertial measurement unit (IMU) coincide here. In the more general case, an additional transformation must be included that accounts for the difference in the coordinate frames of IMU and camera [23].

GPS measurements induce constraints on the vehicle pose, that can be expressed similarly by using the fictitious zero pose s_0

$$f_{gps}(s_0, s_j) = Q_{gps} [R(\Psi_0)^T(t_j - t_0) - t_{gps}] \quad (20)$$

In order to constrain absolute orientation, we employ accelerometers as inclinometers [14]. With g being the gravity vector in the global coordinate frame, this constraint is calculated as

$$f_g(s_0, s_j) = Q_g [R(\Psi_j)^T R(\Psi_0) g - a] \quad (21)$$

where a denotes an accelerometer measurement and Q_g a weight matrix corresponding to a sufficiently large measurement covariance to account for accelerations of the vehicle.

B. Graph Reduction

The requirement of real-time capability in terms of throughput imposes strict constraints on the number of poses that can be considered in the optimization. At the same time, it should be possible to integrate multiple GPS measurements into the optimization to exploit their implications on the global orientation of the vehicle path. In order to address these conflicting objectives, the pose estimation includes a graph reduction scheme, similar to [18], [14], and is carried out over a sliding-window of most recent nodes.

We apply the following steps in order to marginalize out a node s_M , a procedure which is depicted graphically in Fig. 3:

- 1) Select the sub-graph consisting of s_M and all directly connected constraints and nodes. Let $S = [s_0, s_1, \dots, s_n, s_M]$ denote the set of nodes in the sub-graph, including the node s_M , which is later marginalized, and the zero pose s_0 . As inclinometer information is available for every pose, the sub-graph always includes the zero node. Optimize the sub-graph and collect its overall state as well as the overall Jacobian $J = \nabla F(S)$.
- 2) Calculate the approximated Hessian matrix of this estimation as $H = J^T J$. Marginalize out the rows and columns of H and S that correspond to node s_M , using the *Schur complement* [14]. With

$$H = \begin{bmatrix} H_{nn} & H_{nM} \\ H_{nM}^T & H_{MM} \end{bmatrix} \quad (22)$$

where H_{nn} denotes the block of H that does not correspond to any parameter of node s_M , the marginalization is performed as

$$\tilde{H} = H_{nn} - H_{nM} H_{MM}^{-1} H_{nM}^T \quad (23)$$

The resulting matrix corresponds to the inverse covariance matrix for the remaining poses in the coordinate frame of s_0 .

- 3) Select a root node $s_r \in S \setminus \{s_0, s_M\}$, remove the corresponding rows and columns from \tilde{H} and S , and obtain \hat{S} and \hat{H} by transforming the state S and Hessian \tilde{H} into the coordinate system of s_r using the transformation $T(s_r, s_i)$ introduced in Eq. 17 and its Jacobians. Render constraints of the selected sub-graph inactive, connect s_r and s_M through a fixed 6 DoF transformation $T(s_r, s_M)$, and introduce a new constraint with $h(s_r, s_0, \dots, s_n) = [T(s_r, s_0), \dots, T(s_r, s_n)]^T$, $x = \hat{S}$ and $Q^T Q = \hat{H}$ connecting the remaining nodes.

The pose estimation repeatedly marginalizes nodes and rigidly connects the marginalized nodes to remaining active nodes through a 6 DoF transformation. As the marginalization of a node involves a linearization of the sub-system connected to this node, the resulting reduced system is only exact as long as the sub-system is moved rigidly. Otherwise, linearization errors are introduced into the solution. The implications of these errors are twofold: On the one hand, they impose an upper bound on the number of consecutive nodes that can be marginalized without a significant sacrifice in accuracy, which we determined to be about 80 poses for our experiments. On the other hand, we observed that these linearization errors can introduce false constraints on previously unconstrained DoF of the global orientation. For sparse GPS measurements, even these small constraints may result in a serious distortion of the path. In order to overcome this problem, the transformation $T(s_r, s_0)$ of any root node to the zero node has been adapted to only impose constraints on the observable parts of the state given the sensor data fused into this edge of the graph. For improved stability, nodes with attached GPS constraints are not marginalized.

V. RESULTS AND DISCUSSION

We evaluate results of our visual odometry work from Section III and pose optimization algorithm presented in Section IV on real world data collected from a river.

A. Experimental Setup

The dataset used in our experiments was captured on the Allegheny river near Pittsburgh. A sensor suite consisting of a stereo camera pair, an IMU and a consumer-grade L1 GPS was mounted on a tripod and was taken onto a narrow section of the river on a pontoon boat. The boat travelled in a loop of roughly 2 km overall length, which corresponded to about 12 minutes of IMU and GPS measurements and almost 10,000 video frame pairs. In the course of our experiments, we acquired ground truth using a high accuracy L1/L2 Trimble GPS system, with RTKLIB [24] post-processing. We assume the L1/L2 system as too expensive and heavy for the applications we focus on, and we use this just as ground-truth. As input for our pose graph optimization we used a MicroStrain 3DM-GX3-35 inertial measurement unit with integrated L1 GPS, that measured angular velocities and linear accelerations at 100 Hz.

A Point Grey Bumblebee2 stereo camera with an image resolution of 1024 x 768 and 97° horizontal field of view,

provided the imagery to the visual odometry at a rate of 15 Hz.

B. Visual Odometry Results

We implement as a basis of our visual odometry work the algorithm of [2] as a representation of a standard stereo visual odometry approach, that uses RANSAC to remove outliers and Levenberg-Marquardt to solve for motion. We estimate the bias for each frame and correct to provide our motion solution, Eq. 12.

Fig. 4 shows a comparison between the error experienced by the algorithm from [2] and the corrected solution we generate frame by frame. The plots demonstrate that the standard visual odometry underestimates the motion on average by approximately 10%, and that our solution gives an improved estimate at around 2% error. The variance of our estimate scales accordingly.

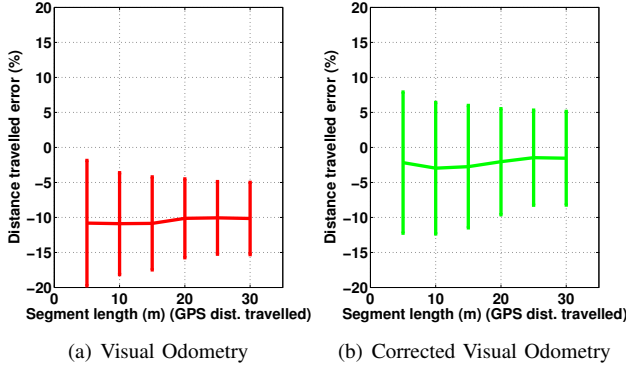


Fig. 4. Mean and standard deviation of errors in stereo visual odometry experienced over a 2km traverse on a river. The output of a standard stereo visual odometry algorithm [2] compared against our corrected solution. To generate this plot, we subdivided the path into segments of the sizes indicated on the x axis and evaluated the displacement of the end position of the accumulated visual odometry to ground truth recorded with a sophisticated L1/L2 GPS system. The standard visual odometry underestimates the motion on average by approximately 10%, while our correction improves the underestimation to about only 2%. The variance scales according to our correction.

C. Global Pose Optimization Results

Next we discuss the results of the global pose optimization. We first quantitatively analyze the errors in the optimization given a very sparse input of GPS data, at a rate of one GPS measurement approximately every two minutes. We compare the optimization both with the standard visual odometry solution and with the corrected solution we derive from Eq. 12 and present in the plot in Fig 5(a). The plot illustrates that in-between the GPS measurements the uncorrected visual odometry solution drifts up to 20m away from the correct solution, and with a mean error of 10m. The corrected solution on the other hand has less drift and at no stage is the error larger than 10m and the mean error is over two times smaller at 5m from ground truth.

Next we visually present the estimated trajectories overlaid on satellite images, in Fig. 6. The figure depicts the reconstructed vehicle path for different levels of GPS availability,

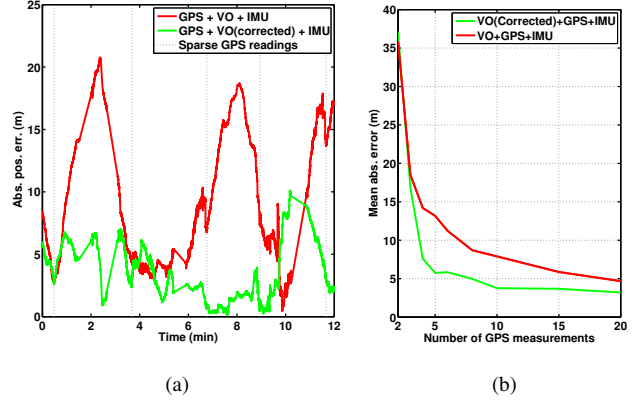


Fig. 5. Visualization of the position error in the estimated poses over a 2km traverse on a river. In Fig. (a), we use a very sparse input of GPS with 6 readings input randomly over the 12 minutes traverse, at an approximate rate of one every two minutes. The red line indicates the solution using a typical visual odometry algorithm [2]. The green line depicts results for the corrected visual odometry provided by Eq. 12. GPS readings used in the pose estimation are indicated by grey vertical lines. Fig. (b) displays the mean error of the global pose estimation using varying number of GPS measurements. The mean error in absolute position was computed over all poses and 10 iterations of the dataset, each iteration with randomly selected GPS measurements. We observed a moderate growth in the error when reducing the number of readings from 20 to 5 followed by a rapid growth in error for less than 5 readings as the drift in visual odometry becomes dominant.

overlaid onto an aerial image of the area. The black path displays the ground truth for the motion, while red is the uncorrected and green the corrected visual odometry. The six black marks indicate the GPS readings used for the global state estimation. With only six measurements over 2km traverse, the global position and orientation of the path could be recovered with accuracy of mean of 5m. The solution using the uncorrected visual odometry is biased and most apparent at the turns in the trajectory where the path is underestimated by approximately 20m, whereas the corrected visual odometry solution replicates the path accurately without a significant bias. The reduced graph of this sequence of about 10k stereo frames contains 168 active nodes and can be solved in about 300 milliseconds.

Finally in Fig. 5(b) we look at the effect on accuracy that introducing varying amount of GPS measurements has on the solution. As the number of GPS measurements is decreased the error in the corrected visual odometry remains low at 6m with even as few as 5 readings over a 2km traverse. The drift in visual odometry is too large to provide an accurate solution with fewer than 5 readings and the error grow dramatically with less GPS readings. By comparison the solution with uncorrected visual odometry is consistently less accurate from 20 GPS readings to a point where at 5 readings the error is twice as large, diverging from truth by 13m on average. With less than 5 readings both corrected and uncorrected solutions diverge substantially from ground truth, where sources of drift other than the underestimation bias dominate. However, even in these extreme cases the pose estimation is capable of inferring a reasonable heading estimate from sparse GPS measurements.

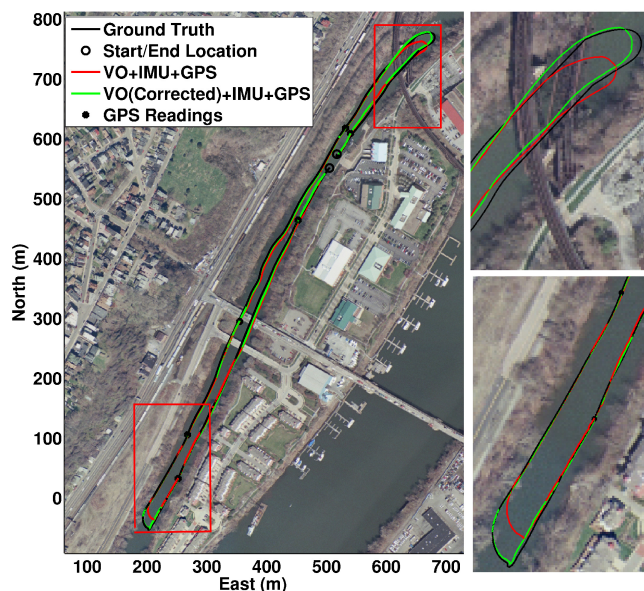


Fig. 6. Visualization of the global pose optimization using six sparse GPS readings over a 12 minutes, 2km traverse. The plot illustrates a comparison between the corrected and uncorrected visual odometry. The black path represent the ground truth of the motion obtained with a sophisticated L1/L2 RTK GPS system. On the right of the figure, close-up views of the path at the southern and northern end of the traverse are depicted. Here the underestimation in the uncorrected visual odometry is apparent as the vehicle turns around to return back along the river with approximately 20m from truth at the apex and the corrected visual odometry solution is within 5m.

VI. CONCLUSION AND FUTURE WORK

Our visual odometry results validate our assumption that the bias for the true camera motion is locally smooth and can be approximately estimated at its initial estimate. Our future work will try to determine if there are conditions under which our assumptions fail to reproduce the true behavior of the bias. We perceive the significance of the motion bias studied in this work as not being limited to riverine environments. In many experiments conducted outdoors, aerial vehicle applications being a prime example, there will be moments where the majority of observed features are at large distance with respect to the stereo baseline and in these cases the proposed bias correction can improve accuracy.

The results of our pose estimation system showed globally consistent paths using very few GPS measurements. We employed a batch optimization method over a sliding window of poses, which required the presence of at least two GPS measurements within the window to infer global heading from relative motion and absolute positions. To keep this computationally tractable for intermittent GPS, poses were continuously marginalized from the system, which inevitably introduced linearization errors. We found that these errors resulted in false constraints on unobservable parts of the state and that special care had to be taken to avoid these from distorting the estimated path when GPS is only intermittently available. Our future work will incorporate kinematic constraints into the graph to help further constrain the state estimate.

REFERENCES

- [1] A. Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer, and S. Singh, "Perception for a river mapping robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011*, September 2011.
- [2] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.
- [3] D. Scaramuzza, "Performance evaluation of 1-point-ransac visual odometry," *Journal of Field Robotics*, vol. 28, no. 5, pp. 792–811, 2011.
- [4] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, p. 169186, 2007.
- [5] K. Konolige and M. Agrawal, "Frame-frame matching for realtime consistent visual mapping," in *IEEE International Conference on Robotics and Automation, 2007*, 2007, pp. 2803–2810.
- [6] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *International Conference on Intelligent Robots and Systems*, 2008, 2008.
- [7] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010*, 2010, pp. 4161–4168.
- [8] J. Kelly, S. Saripalli, and G. Sukhatme, "Combined visual and inertial navigation for an unmanned aerial vehicle," in *Proceedings of the International Conference on Field and Service Robotics*, 2007.
- [9] G. Sibley, L. Matthies, and G. Sukhatme, "Bias reduction filter convergence for long range stereo," in *12th International Symposium of Robotics Research*, 2005.
- [10] G. Sibley, G. Sukhatme, and L. Matthies, "The iterated sigma point kalman filter with applications to long range stereo," in *Robotics: Science and Systems*, 2006, pp. 263–270.
- [11] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [12] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *IEEE International Conference on Robotics and Automation*, 2005, 2005, pp. 2417–2424.
- [13] H. Strasdat, J. Montiel, and A. Davison, "Real-time monocular slam: Why filter?" in *IEEE International Conference on Robotics and Automation*, 2010, 2010, pp. 2657–2664.
- [14] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077, 2008.
- [15] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *18th International Conference on Pattern Recognition*, 2006, vol. 3, 2006, pp. 1063–1068.
- [16] S. Thrun and M. Montemerlo, "The graph slam algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, p. 403, 2006.
- [17] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A General Framework for Graph Optimization," *Journal of Autonomous Robots*, vol. 30, no. 1, pp. 25–39, 2011.
- [18] J. Folkesson and H. Christensen, "Graphical slam for outdoor applications," *Journal of Field Robotics*, vol. 24, pp. 1–2, 2007.
- [19] M. I. Lourakis, "Sparse non-linear least squares optimization for geometric vision," in *European Conference on Computer Vision*, 2010, vol. 2, 2010, pp. 43–56.
- [20] L. Matthies and S. Shafer, "Error modelling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 239–250, 1987.
- [21] K. Konolige, "Large-scale map-making," in *National Conference on Artificial Intelligence*. AAAI Press, 2004, pp. 457–463.
- [22] M. Bryson and S. Sukkarieh, "Building a robust implementation of bearing-only inertial slam for a uav," *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 113–143, 2007.
- [23] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.
- [24] T. Takasu, N. Kubo, and A. Yasuda, "Development, evaluation and application of rtklib: A program library for rtk-gps," in *GPS/GNSS symposium*, 2007.