High-Precision Globally-Referenced Position and Attitude via a Fusion of Visual SLAM, Carrier-Phase-Based GPS, and Inertial Measurements

Daniel P. Shepard and Todd E. Humphreys
The University of Texas at Austin, Austin, TX

***Abstract*—A novel navigation system for obtaining high-precision globally-referenced position and attitude is presented and analyzed. The system is centered on a bundle-adjustment-based visual simultaneous localization and mapping (SLAM) algorithm which incorporates carrier-phase differential GPS (CDGPS) position measurements into the bundle adjustment in addition to measurements of point features identified in a subset of the camera images, referred to as keyframes. To track the motion of the camera in real-time, a navigation filter is employed which utilizes the point feature measurements from all non-keyframes, the point feature positions estimated by bundle adjustment, and inertial measurements. Simulations have shown that the system obtains centimeter-level or better absolute positioning accuracy and sub-degree-level absolute attitude accuracy in open outdoor areas. Moreover, the position and attitude solution only drifts slightly with the distance traveled when the system transitions to a GPS-denied environment (e.g., when the navigation system is carried indoors). A novel technique for initializing the globally-referenced bundle adjustment algorithm is also presented which solves the problem of relating the coordinate systems for position estimates based on two disparate sensors while accounting for the distance between the sensors. Simulation results are presented for the globally-referenced bundle adjustment algorithm which demonstrate its performance in the challenging scenario of walking through a hallway where GPS signals are unavailable.**

I. Introduction

Cameras remain one of the most attractive sensors for motion estimation because of their inherently high information content, low cost, and small size. Visual simultaneous localization and mapping (SLAM) leverages this vast amount of information provided by a camera to estimate the motion of the user and a map of the environment seen by the camera with a high degree of precision as the user moves around the environment. However, the utility of stand-alone visual SLAM is severely limited due to its scale ambiguity (for monocular cameras) and lack of a global reference.

Much prior work in visual SLAM has focused on either eliminating the scale ambiguity, through the inclusion of inertial measurements [1], [2] or GPS carrier-phase measurements [3], or employing previously-mapped visually recognizable markers, referred to as fiduciary markers [4]. In contrast, there has been little prior work that attempts to solve the problem of anchoring the local navigation solution produced by visual SLAM to a global reference frame without the use of an a priori map of the environment, even though the no-prior-map-technique is preferred or required for many applications. The few papers addressing this issue typically employ estimation architectures with performance significantly inferior to an optimal estimator and lean heavily on magnetometers and inertial measurement units (IMUs) for attitude determination, which results in poor attitude precision for all but the highest quality magnetometers and IMUs [5]–[9].

The visual SLAM framework reported in prior literature that comes closest to that reported in this paper is Bryson's visual SLAM algorithm from [10]. Bryson used a combination of monocular visual SLAM, inertial navigation, and GPS to create a map of the terrain from a UAV flying 100 m above the ground. However, this algorithm was incapable of running in real-time; estimation of the map of the environment and vehicle motion was performed after-the-fact. Additionally, the accuracy of the solution in a global sense was severely limited because standard positioning service (SPS) GPS, which is accurate to only a few meters, was used. The inertial measurements and visual SLAM help to tie the GPS measurements together during the batch estimation procedure to increase the accuracy of the resulting solution, but the resulting position estimates were still only accurate to decimeter level.

The visual SLAM framework presented in this paper is inspired by Klein and Murray's Parallel Tracking and Mapping (PTAM) technique [11], which is a stand-alone visual SLAM algorithm that separates tracking of the position and attitude of the camera and mapping of the environment into two separate threads. The mapping thread performs a batch estimation procedure, referred to as bundle adjustment, that operates on a subset of the camera images, referred to as keyframes, that are chosen for their spatial diversity to estimate the position of identified point features and the position and orientation of the camera at each keyframe. The tracking thread identifies point features in each frame received from the camera and determines the current position and attitude of the camera using the point feature measurements from the current frame and the current best estimate of the positions of the point features from the mapping thread.

Like PTAM, the visual SLAM framework presented in this paper separates tracking of the position and attitude of the camera and mapping of the environment into two separate threads. However, the mapping thread's bundle adjustment algorithm presented in this paper additionally employs carrier-phase differential GPS (CDGPS) position estimates, interpo-

lated to the time the keyframes were taken. This allows the mapping thread to determine the position of each point feature and the position and attitude of the camera at each keyframe to high-precision in a global coordinate system without the use of a magnetometer or IMU and without an a priori map of the environment. When CDGPS position estimates are not available (e.g., when the navigation system is carried indoors), the mapping thread continues to operate without these measurements at new keyframes. However, the accuracy of the estimates of any newly identified point features, and thus the position and attitude of the system, decays slowly with the distance traveled in the GPS-denied environment. Since information about previous keyframes is maintained, returning to a previously-visited area in the GPS-denied environment will aid in fixing up any accumulated errors, a condition referred to as loop closure.

The tracking thread for the visual SLAM framework presented in this paper maintains the point feature identification functionality of PTAM's tracking thread but incorporates a navigation filter. This filter greatly improves the accuracy of the best estimate of the current position and attitude of the camera by providing a better motion model, through the incorporation of IMU measurements, and utilizing information obtained from all non-keyframes. The filter additionally improves the robustness and computational efficiency of the tracking and mapping threads by aiding in recovery during rough dynamics, reducing the search space for feature identification, and reducing the number of required batch iterations for the mapping thread.

One significant advantage of this navigation system over other high-precision navigation systems is that it can be implemented using inexpensive sensors. Modern digital cameras are inexpensive, high-information-content sensors. Inexpensive GPS receivers are available today that produce the single-frequency carrier-phase and psuedorange measurements required to determine a CDGPS position solution. An inexpensive IMU can also be employed because the navigation system does not rely on the IMU for long-term state propagation or attitude determination.

One promising application for this type of navigation system is augmented reality. Augmented reality (AR) is a concept closely related to virtual reality (VR), but has a fundamentally different goal. Instead of replacing the real world with a virtual one like VR does, AR seeks to produce a blended version of the real world and context-relevant virtual elements that enhance or augment the user's experience in some way, typically through visuals. The relation of AR to VR is best explained by imagining a continuum of perception with the real world on one end and VR on the other. On this continuum, AR would be placed in between the real world and VR with the exact placement depending on the goal of the particular application of AR.

The primary limiting factor for AR is the fact that AR requires extremely precise navigation to maintain the illusion of realism of the virtual objects augmented onto the view of the real world. AR applications simply fail to impress without this illusion of realism, and the human eye is fairly good at picking up on this. Additionally, large errors in the registration of virtual objects (i.e., position and orientation of virtual objects relative to the real world) make it impossible for a user to interact with these objects.

Many current successful AR applications rely on visual SLAM for relative navigation, which results in accuracy suitable for many applications. However, there are many AR applications, such as construction, utility work, social networking, and multiplayer games, that are awkward or impossible to do using relative navigation alone because of the need to relate navigation information in a consistent coordinate system. The navigation system presented in this paper has the required precision in a global reference frame to serve as a viable AR platform for all of these applications.

This paper begins with a discussion of the estimation architecture that details the differences between the proposed estimation architecture and that of the optimal estimator. Next, the state vectors and measurement models for the bundle adjustment and navigation filter are described. Then, the bundle adjustment algorithm is detailed including a novel technique for initialization of the globally-referenced bundle adjustment. This is followed by a detailed description of the navigation filter. Finally, results from simulations of the bundle adjustment are presented.

II. Estimation Architecture

The eventual goal of the work presented in this paper is the creation of a high-precision globally-referenced navigation system based on a fusion of visual SLAM, CDGPS, and inertial measurements that is capable of operating in real-time. An important consideration for any multi-sensor navigation system is how the information from these sensors will be combined to estimate the state. An optimal estimator would, by definition, attain the highest precision state estimate for any given set of measurements, but operation of an optimal estimator in real-time is impractical for this system due to finite computational resources and the high computational demand of visual SLAM. Therefore, compromises must be made with regard to the optimality of the estimator to enable real-time operation, as is typically the case. The remainder of this section details the compromises made to enable real-time performance by describing the differences between the optimal estimator and the estimation architecture proposed in this paper. Note that the optimal estimator and the intermediate architectures leading to the final proposed architecture are only notional and are used simply to draw a comparison with the final proposed architecture.

A. Optimal Estimator

To highlight the differences between the proposed estimation architecture and the optimal estimator, the optimal estimator for this problem must first be presented. Due to non-linearities, the optimal estimator, in a least-squares sense, requires that the measurements from all sensors at all time epochs received thus far be processed together in a single least-squares batch estimator. Before introducing the state vector for this batch estimator, it is necessary to define what measurements from each sensor the estimator employs, since this choice may alter the state. For example, there are three types of measurements that can be taken from a GPS receiver which represent different stages in the processing and, for CDGPS, will change the state of the estimator depending on which of these types of measurements is employed. When coupling GPS measurements with those from another navigation sensor, it is conventional

to consider three levels of coupling based on the types of GPS measurements used by the estimator and the details of the estimation architecture. These levels of coupling and the types of GPS measurements associated with those levels of coupling are as follows

1) Loosely-coupled, which uses position and time estimates
2) Tightly-coupled, which uses pseudorange and carrier-phase measurements for each GPS signal
3) Ultra-tightly-coupled, which uses in-phase and quadrature accumulations for each correlator tap and GPS signal

As a general rule-of-thumb, more tightly coupled estimation architectures will result in better performance. Therefore, the optimal estimator considered here employs an ultra-tightly-coupled architecture where the tracking loops of both GPS receivers, a reference receiver at a known location and the mobile receiver, are driven by the estimate of the state (i.e., a vector tracking loop). The sensor measurements used in this estimator are (1) the in-phase and quadrature accumulations for each GPS signal for the prompt, early, and late correlator taps from both GPS receivers, reference and mobile, (2) the image feature coordinates in each image, and (3) the specific force and angular velocity measurements from the IMU. The state vector for this estimator would include the following

1) Camera poses (i.e., position and attitude of the camera) for each image
2) The velocity of the camera for each image
3) The local clock offset and offset rate from GPS time for both receivers at the time each image was taken
4) The image feature positions
5) Either the accelerometer and gyro biases at each image or the coefficients of a piece-wise polynomial model for the accelerometer and gyro biases
6) The integer ambiguities on the double-differenced carrier-phase measurements, which are formed based on the prompt tap in-phase and quadrature accumulations from both receivers

B. Removal of Inertial Measurements

The first compromise made to reduce the required computational expense of the estimator was to remove the IMU measurements from the batch estimator and the accelerometer and gyro biases from the state of the batch estimator. Due to the already high-precision of the GPS and vision measurements, the measurements from the IMU do not significantly contribute to the accuracy of the state estimate and are not necessary for observability of the state. The vision measurements act as an extremely high quality IMU by relating the poses of each image and allow for determination of attitude to a high-precision, even without the IMU. Additionally, it is awkward and computationally burdensome to deal with the IMU biases in the batch estimator. Thus, incorporating the IMU measurements into the batch estimator is simply not worth the marginal benefits gained. However, the IMU measurements are useful for propagating the state between frames, which can be performed external to the batch estimator.

C. Scalar GPS Tracking Loops

While the vector GPS tracking loops in an ultra-tightly-coupled architecture do significantly improve robustness of signal tracking and acquisition, this benefit comes at an extremely high price in terms of the computational requirements of the estimator because the estimator has to update the state at a much faster rate to drive the tracking loops. Additionally, the incorporation of visual SLAM results in extremely slow drift in the state estimate during GPS outages and aids in detecting and eliminating carrier-phase cycle slips, which minimizes the impact a vector tracking loop would have on performance over scalar tracking loops. Therefore, a tightly-coupled architecture for the estimator, where the in-phase and quadrature accumulations are replaced with pseudorange and carrier-phase measurements, can be employed instead of an ultra-tightly-coupled architecture with little loss in performance. The local clock offset and offset rate from GPS time for both GPS receivers can also be removed from the state vector. These parameters were necessary in the ultra-tightly-coupled architecture because the vector tracking loops needed an estimate of time, but the tightly-coupled architecture does not require these parameters, so long as the pseudorange and carrier-phase measurements for both receivers are aligned in time to at least GPS standard positioning service (SPS) accuracy. This is a result of the effects of the local clock canceling out when forming the double-differenced pseudorange and carrier-phase measurements in the CDGPS algorithm [12].

D. Separation of CDGPS Processing and Batch Estimator

The third compromise was the separation of CDGPS position estimation from the batch estimator, which was done primarily to simplify the design of the batch estimator. In this loosely-coupled architecture, the CDGPS-based position estimates are incorporated into the batch estimator instead of the pseudorange and carrier-phase measurements, and the double-differenced carrier-phase integer ambiguities are removed from the state vector of the batch estimator. To better understand the effects of removing the CDGPS algorithm from the batch estimator, there are two pertinent questions one must ask:

1) How is the estimation of the double-differenced carrier-phase integer ambiguities affected by incorporation into the batch estimator?
2) How much does the accuracy of the state estimate change when double-differenced carrier-phase measurements with resolved integer ambiguities are incorporated into the batch estimator instead of only the position estimates suggested by those measurements?

As for the first question, it is well understood in literature on CDGPS that the addition of any constraints significantly aids the resolution of the integer ambiguities [13]. The vision measurements are able to constrain pose in a local coordinate system, which will greatly aid in resolving the integer ambiguities (especially during motion). As for the second question, it is difficult to say how much the state estimate will improve without implementing this approach. However, the accuracy of the state estimate is certain to improve at least somewhat. As opposed to the removal of inertial measurements and transition to a tightly-coupled architecture discussed previously, the separation of CDGPS position estimation may have a significant

effect on the performance of the estimator with little change in the computational burden, provided that the integer ambiguities are fixed after convergence rather than continually estimated. Therefore, the authors hope to later reincorporate the CDGPS algorithm into the batch estimator, which is not a trivial matter.

E. Hybrid Batch/Sequential Approach

Computing global solutions for the keyframe poses and image feature positions in the batch estimator is a computationally intensive process that would require immense computational resources if it were performed at the frame-rate of the camera. Therefore, an approach to this problem employing a single batch estimator performing these global solutions for each frame is impractical for real-time applications. Thankfully, the highest accuracy for visual SLAM is obtained by employing a large number of image features, which results in a sparse structure that can be exploited by the batch estimator, and a smaller number of geographically-diverse images, as was shown in [14]. In other words, not all frames are created equal. The principle of diminishing return applies to frames taken from nearly the same camera pose. This means that there is no need to process most of the images in the batch estimator because incorporating these images into the batch estimator will result in little improvement in the accuracy of the global solution. Therefore, new frames should be incorporated into the batch estimator only when the frame to be incorporated is, in some sense, geographically diverse from the other frames already incorporated into the batch estimator. These geographically-diverse frames are referred to as keyframes and can be chosen based on a comparison of prior keyframe poses and an estimate of the current camera pose. As the system moves around, eventually the number of keyframes will become too large to perform global solutions in a reasonable amount of time. At this point, the batch estimator can shed old keyframes and point features that no longer contribute significantly to the estimate of the point feature locations near the system's current position and save the image feature measurements for these keyframes for later use if the system returns to that area.

To be useful as a real-time navigation system, however, the system must maintain a highly accurate estimate of the current pose of the camera. This goal is at odds with the compromise that only select frames be incorporated into the batch estimator. While the non-keyframes do not contribute significantly to the accuracy of the global solution, they do contain valuable information about the pose of the camera at the time they were taken. To account for this, a second estimator can be employed that takes as measurements the non-keyframes and the inertial measurements from the IMU, which aid in propagating the camera pose between frames. The second estimator additionally utilizes the estimates of the image feature locations from the first estimator to tie its navigation solution to the global map. This approach results in a federated estimation architecture where one estimator, the batch estimator discussed in the previous paragraph, is tasked with mapping the environment and a second estimator is tasked with tracking the motion of the camera. This is similar to the approach taken by Klein and Murray's stand-alone visual SLAM algorithm called PTAM [11].

One issue with using this federated approach for the estimator, which is discussed in further detail in section VI, is that the second estimator cannot maintain cross-covariances between the image feature positions from the batch estimator without destroying its ability to operate in real-time. This means the second estimator cannot hope to truly maintain a consistent estimate of the state covariance and an ad-hoc inflation of the covariance matrix must be used. However, these cross-covariances between image feature positions will typically be small in practice, which is demonstrated by the simulation results in Sec. VII.

The system reported in this paper employs a sequential estimator or filter as this second estimator. Thus, the estimation architecture presented in this paper describes a hybrid batch/sequential estimator. One could instead employ a batch estimator that uses only the last few frames, but incorporating the IMU measurements into this framework would be somewhat awkward. However, this is a viable option that might be explored in future work.

F. Proposed Estimation Architecture

Based on the previous discussion, an estimation architecture was designed that has the potential for real-time operation. A block diagram of this estimation architecture is shown in Fig. 1. The blocks on the far left of Fig. 1 are all the sensors for the system; camera, reference GPS receiver, mobile GPS receiver, and IMU.

The most important components of this estimation architecture are the two blocks on the far right of Fig. 1. The upper block is the batch estimator responsible for creating a high-precision globally-referenced map of the environment based on image feature measurements from the keyframes and CDGPS-based position estimates, when available, interpolated to the time the keyframes were taken. This process of estimating a map of the environment based on keyframes in a batch estimator is commonly referred to as bundle adjustment. In this particular case, the bundle adjustment is augmented with CDGPS-based position estimates which anchor the bundle adjustment to a global coordinate system. The lower block is the sequential estimator or navigation filter responsible for maintaining an accurate estimate of the current pose of the camera based on image feature measurements from non-keyframes, IMU specific force and angular velocity measurements, and the image feature position estimates from the bundle adjustment. In addition to being the primary output of the estimator as a whole, the camera pose estimated by the navigation filter is also used in several other components of the estimator to aid in improving computational efficiency and performance. These two components of the estimator, bundle adjustment and navigation filter, are the main focus of the remainder of this paper.

In addition to the bundle adjustment and the navigation filter, there are several other components of the estimation architecture that are responsible for producing the measurements that are later input to the bundle adjustment and the navigation filter. The CDGPS filter, shown in the middle of Fig. 1, is responsible for estimating the position of the GPS antenna and the double-differenced carrier-phase integer ambiguities based on the pseudorange and carrier-phase measurements from the reference and mobile GPS receivers. An estimate of the current camera pose from the navigation filter is provided to the

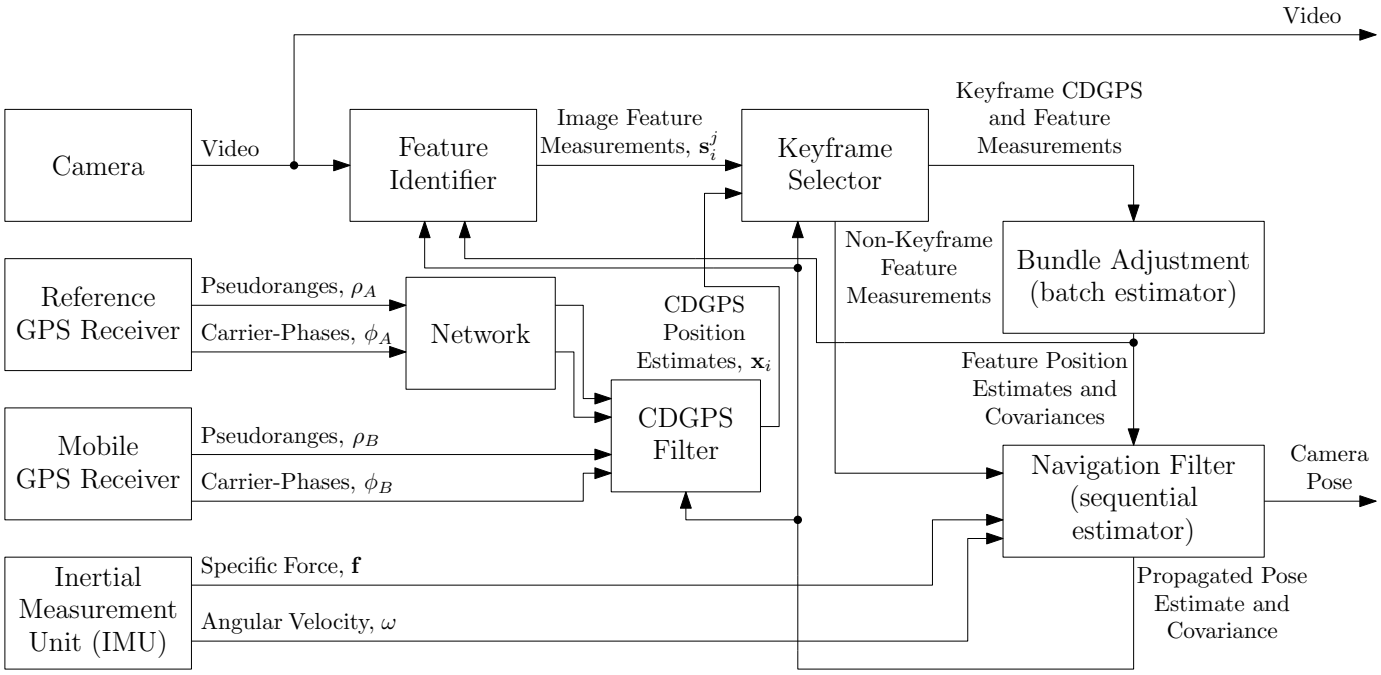

Fig. 1. A block diagram of the proposed estimation architecture.

CDGPS filter only for linearization and, thus, does not create correlation between the CDGPS-based position estimates and the navigation filter's pose estimate. This is important because any correlation between the output of the navigation filter and the input to the bundle adjustment would destroy the consistency of the map created by the bundle adjustment and could cause divergence of the estimator. The feature identifier, shown to the right of the camera in Fig. 1, finds and matches features in each image received from the camera. To reduce the computational burden of the feature identifier, estimates of the camera pose and image feature positions are provided to the feature identifier by the navigation filter and bundle adjustment, respectively, which reduces the search space for each image feature. Although not shown explicitly in Fig. 1, the feature identifier also identifies new features matched between multiple keyframes. The keyframe selector, shown to the right of the feature identifier in Fig. 1, employs a set of heuristics that determines whether a frame is geographically diverse enough, relative to the current keyframes, to be considered a new keyframe. These heuristics are based on the estimate of the current camera pose provided by the navigation filter. While these components (CDGPS filter, feature identifier, and keyframe selector) serve important roles in this estimation architecture, they will only be discussed superficially throughout the remainder of this paper because this is not where the contributions of this paper lie and there is already a plethora of literature on these components individually.

III. State Vector

Before discussing the measurement models and estimation algorithms for the bundle adjustment and the navigation filter, it is appropriate to first introduce the state vectors for each estimator.

A. Bundle Adjustment State

The bundle adjustment is responsible for producing a globally-referenced map of the environment and, as such, must include the position of each image feature in the global coordinate system in its state vector. As a byproduct of producing this map, the camera poses at each keyframe must also be estimated in this global coordinate system. Therefore, the state vector for the bundle adjustment is as follows

$$\mathbf{X}_{BA} = \begin{bmatrix} \left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_1}\right)^T & \left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}_1}\right)^T & \cdots & \left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_N}\right)^T & \left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}_N}\right)^T, \\ \left(\mathbf{x}_{\mathcal{G}}^{p_1}\right)^T & \cdots & \left(\mathbf{x}_{\mathcal{G}}^{p_M}\right)^T \end{bmatrix}^T \tag{1}$$

where $\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_i}$ is the position of the camera at the ith keyframe in the global coordinate system, $\mathbf{q}_{\mathcal{G}}^{\mathcal{C}_i}$ is the quaternion representation of the attitude of the camera at the ith keyframe relative to the global coordinate system, N is the number of keyframes in the bundle adjustment, $\mathbf{x}_{\mathcal{G}}^{p_j}$ is the position of the jth image feature in the global coordinate system, and M is the number of image features in the map.

The $\mathcal{C}_i$ frame is the camera frame at the time the ith keyframe was taken. The camera frame, which will be denoted as $\mathcal{C}$, is defined as the reference frame centered on the camera lens with the z-axis pointing down the bore-sight of the camera, the x-axis pointing to the right, and the y-axis pointing down to complete the right-handed triad. The $\mathcal{G}$ frame is the Earth-Centered Earth-Fixed (ECEF) coordinate system. Note that for any attitude representation in this paper $(\cdot)_{\mathcal{B}}^{\mathcal{A}}$ represents a rotation from the $\mathcal{A}$ frame to the $\mathcal{B}$ frame.

B. Navigation Filter State

The navigation filter is responsible for maintaining an estimate of the current camera pose, which must be included in the state vector. As part of maintaining an estimate of the current camera pose, the camera pose after each measurement update must be propagated forward in time to the next measurement. This is accomplished through the use of accelerometer and gyro measurements which include bias terms that must be estimated. Therefore, the state vector for the navigation filter is as follows

$$\mathbf{X}_F = \begin{bmatrix} \left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}\right)^T & \left(\mathbf{v}_{\mathcal{G}}^{\mathcal{C}}\right)^T & \left(\mathbf{b}_{\mathcal{B}}^{f}\right)^T & \left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)^T & \left(\mathbf{b}_{\mathcal{B}}^{\omega}\right)^T \end{bmatrix}^T \quad (2)$$

where $\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}$ and $\mathbf{v}_{\mathcal{G}}^{\mathcal{C}}$ are the current position and velocity of the camera in the global coordinate system, $\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}$ is the quaternion representation of the current attitude of the camera relative to the global coordinate system, and $\mathbf{b}_{\mathcal{B}}^{f}$ and $\mathbf{b}_{\mathcal{B}}^{\omega}$ are the current accelerometer and gyro biases, respectively.

The accelerometer and gyro biases are expressed in the $\mathcal{B}$ frame, which is the IMU's reference frame. The transform between the $\mathcal{B}$ frame and the $\mathcal{C}$ frame, which are both body-fixed coordinate systems, is assumed to be fixed and either measured or calibrated ahead of time. This transformation is given by

$$(\cdot)_{\mathcal{C}} = R_{\mathcal{C}}^{\mathcal{B}}(\cdot)_{\mathcal{B}} - \mathbf{x}_{\mathcal{C}}^{\mathcal{B}} \quad (3)$$

where $R_{\mathcal{C}}^{\mathcal{B}}$ is the rotation matrix relating the two coordinate systems and $\mathbf{x}_{\mathcal{C}}^{\mathcal{B}}$ is the location of the IMU in the $\mathcal{C}$ frame. While this transformation could be estimated on-the-fly instead of simply measured or calibrated ahead of time, the transformation is only weakly observable and does not need to be known with great precision, since the estimator does not rely heavily on the accuracy of this transformation.

IV. Measurement Models

This section presents the measurement models employed by both the bundle adjustment and the navigation filter. As a matter of notation for this paper, parameters, when substituted into models, will be denoted with either a bar, $\bar{(\cdot)}$, for a priori estimates or a hat, $\hat{(\cdot)}$, for a posteriori estimates. Any parameter without these accents is the true value of that parameter. When a state vector or an element of a state vector has a delta in front of it, $\delta(\cdot)$, this represents a linearized correction term to the current value of that state variable. The same accent rules also apply to delta states.

Before presenting the measurement models, it is appropriate to discuss how the quaternions representing the attitude of the camera will be handled within the estimator. Quaternions are a non-minimal attitude representation that is constrained to have unit norm. To enforce this constraint, the quaternion elements of the state are replaced in the state with a minimal attitude representation, generally denoted as $\delta\mathbf{e}$, during measurement updates and state propagation [15]. This is accomplished through the use of differential quaternions, which represent a small rotation from the current attitude to give an updated estimate of the attitude through the equation

$$\mathbf{q}' = \delta\mathbf{q}(\delta\mathbf{e}) \otimes \mathbf{q} \quad (4)$$

where $\mathbf{q}'$ is the updated attitude estimate, $\otimes$ represents quaternion multiplication, and $\delta\mathbf{q}(\delta\mathbf{e})$ is the differential quaternion, which is closely approximated as follows

$$\begin{aligned} \delta\mathbf{q}(\delta\mathbf{e}) &= \begin{bmatrix} \hat{\mathbf{e}}\sin\left(\frac{\delta\theta}{2}\right) \\ \cos\left(\frac{\delta\theta}{2}\right) \end{bmatrix} \\ &\approx \begin{bmatrix} \hat{\mathbf{e}}\frac{\delta\theta}{2} \\ \sqrt{1-\left|\left|\hat{\mathbf{e}}\frac{\delta\theta}{2}\right|\right|^2} \end{bmatrix} = \begin{bmatrix} \delta\mathbf{e} \\ \sqrt{1-||\delta\mathbf{e}||^2} \end{bmatrix} \end{aligned} \quad (5)$$

where $\cos\left(\frac{\delta\theta}{2}\right)$ is approximated as $\sqrt{1-||\delta\mathbf{e}||^2}$ instead of the typical 1 to comply with the quaternion constraint. This approximation allows for reduction of the quaternion to a minimal three-element representation, $\delta\mathbf{e}$, and is useful for preserving the quaternion constraint in an estimator, as shown in [15]. During initial convergence of an estimator, the assumption that $\delta\theta$ is small may be violated and could cause $\sqrt{1-||\delta\mathbf{e}||^2}$ to become imaginary. To protect against this scenario, a less accurate form of the differential quaternion is used whenever $||\delta\mathbf{e}||^2 > 1$. This form of the differential quaternion is

$$\delta\mathbf{q}(\delta\mathbf{e}) = \frac{1}{\sqrt{1+||\delta\mathbf{e}||^2}} \begin{bmatrix} \delta\mathbf{e} \\ 1 \end{bmatrix} \quad (6)$$

This completely specifies the multiplicative update to the quaternion. All other states are updated in the typical additive fashion

$$(\cdot)' = (\cdot) + \delta(\cdot) \quad (7)$$

A. CDGPS Position Measurements

The CDGPS filter provides estimates of the position of the GPS antenna that are accurate to within a couple centimeters. One important observation is that these are not estimates of the position of the camera lens. Therefore, the position of the GPS antenna relative to the camera lens, which is assumed to be fixed and measured or calibrated ahead of time, must be taken into account. Unlike the transformation between the $\mathcal{B}$ frame and the $\mathcal{C}$ frame described previously, it is of prime importance that the position of the GPS antenna in the $\mathcal{C}$ frame be known as accurately as possible because any errors in this parameter directly translates to errors in the estimated state. Thankfully, this parameter is observable provided that the system is rotated at least somewhat in all directions. If it is not possible to measure this vector to at least millimeter accuracy, then a calibration procedure can be defined using the bundle adjustment approach presented in this paper with a state vector that is augmented with the position of the GPS antenna in the $\mathcal{C}$ frame.

These CDGPS position estimates can be modeled as follows

$$\mathbf{x}_{\mathcal{G}}^{A} = \mathbf{h}_x\left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}, \mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right) + \mathbf{w}_x = \mathbf{x}_{\mathcal{G}}^{\mathcal{C}} + R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{A} + \mathbf{w}_x \tag{8}$$

where $\mathbf{h}_x(\cdot)$ is the non-linear measurement model for the CDGPS position estimates, $R(\cdot)$ is the rotation matrix corresponding to the argument, $\mathbf{x}_{\mathcal{C}}^{A}$ is the position of the GPS antenna in the $\mathcal{C}$ frame, and $\mathbf{w}_x$ is zero-mean Gaussian white noise with covariance matrix given by the CDGPS filter. The non-zero partial derivatives of this model with respect to the state variables are

$$\left.\frac{\partial \mathbf{h}_x\left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}, \mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)}{\partial \mathbf{x}_{\mathcal{G}}^{\mathcal{C}}}\right|_{\bar{\mathbf{X}}} = I \tag{9}$$

$$\left.\frac{\partial \mathbf{h}_x\left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}, \mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)}{\partial \delta\mathbf{e}_{\mathcal{G}}^{\mathcal{C}}}\right|_{\bar{\mathbf{X}}} = 2\left[\left(R\left(\bar{\mathbf{q}}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{A}\right)\times\right] \tag{10}$$

where I is the identity matrix and $[(\cdot)\times]$ is the cross product equivalent matrix of the argument defined as

$$[\mathbf{x}\times] = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{11}$$

where x_i is the ith element of $\mathbf{x}$.

B. Image Feature Measurements

To simplify the model for the image feature measurements, it is assumed that a calibrated camera is used and that any distortion caused by the lens is removed by passing the raw measurements through the inverted distortion model prior to passing the measurements to the estimators. This allows the bundle adjustment and the navigation filter to be ambivalent to the distortion model used for the camera. For the simulations presented in Sec. VII, the field-of-view (FOV) model for a fish-eye lens from [16] was employed using parameters calibrated from a real camera using the calibration procedure reported in [11]. The primary effect of using the distortion model in the simulations is to properly model the error covariances on the measurements.

Once the lens distortions have been removed from the raw measurements, what remains is the result of a perspective projection. A perspective projection, also known as a central projection, projects a view of a three-dimensional scene onto an image plane normal to the camera bore-sight located 1 unit in front of the camera through rays connecting three-dimensional locations and a center of projection. A perspective projection can be expressed mathematically as

$$\mathbf{s}_{\mathcal{I}}^{p_j} = \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right) + \mathbf{w}_{p_j} = \begin{bmatrix} \frac{x_{\mathcal{C}}^{p_j}}{z_{\mathcal{C}}^{p_j}} & \frac{y_{\mathcal{C}}^{p_j}}{z_{\mathcal{C}}^{p_j}} \end{bmatrix}^T + \mathbf{w}_{p_j} \tag{12}$$

where $\mathbf{s}_{\mathcal{I}}^{p_j}$ is the distortion-free image feature measurement for the jth feature, $\mathbf{h}_s(\cdot)$ is the perspective projection function, $\mathbf{w}_{p_j}$ is zero-mean Gaussian white noise with covariance matrix given by the feature identifier, and $\mathbf{x}_{\mathcal{C}}^{p_j}$ is related to the state variables through the equation

$$\mathbf{x}_{\mathcal{C}}^{p_j} = \begin{bmatrix} x_{\mathcal{C}}^{p_j} \\ y_{\mathcal{C}}^{p_j} \\ z_{\mathcal{C}}^{p_j} \end{bmatrix} = \left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\right)^T\left(\mathbf{x}_{\mathcal{G}}^{p_j} - \mathbf{x}_{\mathcal{G}}^{\mathcal{C}}\right) \tag{13}$$

The non-zero partial derivatives of this model with respect to the state variables are

$$\left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \mathbf{x}_{\mathcal{G}}^{\mathcal{C}}}\right|_{\bar{\mathbf{X}}} = -\left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \mathbf{x}_{\mathcal{C}}^{p_j}}\right|_{\bar{\mathbf{X}}}\left(R\left(\bar{\mathbf{q}}_{\mathcal{G}}^{\mathcal{C}}\right)\right)^T \tag{14}$$

$$\left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \delta\mathbf{e}_{\mathcal{G}}^{\mathcal{C}}}\right|_{\bar{\mathbf{X}}} = -2\left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \mathbf{x}_{\mathcal{C}}^{p_j}}\right|_{\bar{\mathbf{X}}}\left(R\left(\bar{\mathbf{q}}_{\mathcal{G}}^{\mathcal{C}}\right)\right)^T\left[\left(\bar{\mathbf{x}}_{\mathcal{G}}^{p_j} - \bar{\mathbf{x}}_{\mathcal{G}}^{\mathcal{C}}\right)\times\right] \tag{15}$$

$$\left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \mathbf{x}_{\mathcal{G}}^{p_j}}\right|_{\bar{\mathbf{X}}} = \left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \mathbf{x}_{\mathcal{C}}^{p_j}}\right|_{\bar{\mathbf{X}}}\left(R\left(\bar{\mathbf{q}}_{\mathcal{G}}^{\mathcal{C}}\right)\right)^T \tag{16}$$

where $\left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \mathbf{x}_{\mathcal{C}}^{p_j}}\right|_{\bar{\mathbf{X}}}$ is given by

$$\left.\frac{\partial \mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}}^{p_j}\right)}{\partial \mathbf{x}_{\mathcal{C}}^{p_j}}\right|_{\bar{\mathbf{X}}} = \begin{bmatrix} \frac{1}{\bar{z}_{\mathcal{C}}^{p_j}} & 0 & \frac{-\bar{x}_{\mathcal{C}}^{p_j}}{\left(\bar{z}_{\mathcal{C}}^{p_j}\right)^2} \\ 0 & \frac{1}{\bar{z}_{\mathcal{C}}^{p_j}} & \frac{-\bar{y}_{\mathcal{C}}^{p_j}}{\left(\bar{z}_{\mathcal{C}}^{p_j}\right)^2} \end{bmatrix} \tag{17}$$

C. Inertial Measurements

The inertial measurements consist of 3-axis accelerometer measurements and 3-axis gyro measurements and are used in the navigation filter to aid in propagating the state forward in time. The measurement models presented in this section simply model the relations between these measurements and the acceleration and angular velocity of the IMU with respect to the IMU frame using state variables and should not be interpreted as modeling the dynamics of the state. Filter state dynamics models will be presented in Sec. VI-B.

1) Accelerometer Measurements: A subtle, but extremely important, point regarding accelerometers is that they measure the specific force they experience and not the acceleration. This means that accelerometer measurements include gravitational acceleration, which must be subtracted out. A walking bias term, which was included in the filter's state vector, is also present in these measurements and is typically modeled by the first-order Gauss-Markov process

$$\dot{\mathbf{b}}_{\mathcal{B}}^{f} = \boldsymbol{\nu}_2^f \tag{18}$$

where $\boldsymbol{\nu}_2^f$ is zero-mean Gaussian white noise with a diagonal covariance matrix, $\sigma_{f_2}^2 I$. The covariance of $\boldsymbol{\nu}_2^f$ can be obtained from the IMU specifications. The acceleration of the IMU with respect to the IMU frame can be modeled as

$$\mathbf{a}_{\mathcal{G}}^{\mathcal{B}} = R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)R_{\mathcal{C}}^{\mathcal{B}}\left(\mathbf{f}_{\mathcal{B}} - \mathbf{b}_{\mathcal{B}}^{f}\right) - \frac{G_E}{\left|\left|\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}\right|\right|^3}\mathbf{x}_{\mathcal{G}}^{\mathcal{C}} + \boldsymbol{\nu}_1^f \tag{19}$$

where $\mathbf{f}_{\mathcal{B}}$ is the accelerometer measurement, G_E is the gravitational constant of Earth, and $\boldsymbol{\nu}_1^f$ is zero-mean Gaussian white noise with a diagonal covariance matrix, $\sigma_{f_1}^2 I$. The covariance of $\boldsymbol{\nu}_1^f$ can be obtained from the IMU specifications. Note that using $\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}$ in the gravity term is an approximation to $\mathbf{x}_{\mathcal{G}}^{\mathcal{C}} + R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{\mathcal{B}}$, but the term $R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{\mathcal{B}}$ is extremely small compared to $\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}$ and, thus, negligible.

2) Gyro Measurements: Like the accelerometer measurements, the gyro measurements have a walking bias term, which was included in the filter's state vector. This bias term is also typically modeled by the first-order Gauss-Markov process

$$\dot{\mathbf{b}}_{\mathcal{B}}^{\omega} = \boldsymbol{\nu}_2^{\omega} \tag{20}$$

where $\boldsymbol{\nu}_2^{\omega}$ is zero-mean Gaussian white noise with a diagonal covariance matrix, $\sigma_{\omega_2}^2 I$. The covariance of $\boldsymbol{\nu}_2^{\omega}$ can be obtained from the IMU specifications. The angular velocity of the IMU, which is also the angular velocity of the camera, can be modeled as

$$\boldsymbol{\omega}_{\mathcal{G}}^{\mathcal{C}} = R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right) R_{\mathcal{C}}^{\mathcal{B}}\left(\boldsymbol{\omega}_{\mathcal{B}} - \mathbf{b}_{\mathcal{B}}^{\omega}\right) + \boldsymbol{\nu}_1^{\omega} \tag{21}$$

where $\boldsymbol{\omega}_{\mathcal{B}}$ is the gyro measurement and $\boldsymbol{\nu}_1^{\omega}$ is zero-mean Gaussian white noise with a diagonal covariance matrix, $\sigma_{\omega_1}^2 I$. The covariance of $\boldsymbol{\nu}_1^{\omega}$ can be obtained from the IMU specifications.

V. Bundle Adjustment

Bundle adjustment is a batch estimation procedure employed by many visual SLAM algorithms that takes advantage of the inherent sparsity of the visual SLAM problem. Due to exploiting this sparse structure, the computational complexity of bundle adjustment is linear in the number of image features and cubic in the number of images included in the bundle adjustment. Compared to a sequential estimator, which has computational complexity that is cubic in the number of image features and linear in the number of images, this is a significant improvement because the highest accuracy per CPU cycle for the visual SLAM problem comes from including more image features rather than images [14]. In other words, the compromise of reducing the number of frames incorporated into the bundle adjustment to maintain real-time operation is preferred to reducing the number of image features, which is required for a sequential estimator to operate in real-time. As discussed in Sec. II-E, the frames included in the bundle adjustment are selected based on their geographic diversity and are referred to as keyframes.

The bundle adjustment algorithm presented in this section also incorporates CDGPS position estimates interpolated to the keyframes, which serve to anchor the bundle adjustment solution to a global reference frame. For ease of notation, the equations presented in this section will assume that every keyframe has an associated CDGPS position estimate and every image feature is seen in each keyframe. Obviously, this is not a requirement of the bundle adjustment, and the terms in these equations corresponding to any non-existent measurements are simply ignored by the bundle adjustment. For observability of the globally-referenced visual SLAM problem, four non-coplanar image features seen in three keyframes, that have corresponding globally-referenced position estimates, is sufficient provided that the camera positions for these three keyframes are not collinear [12].

This section begins by defining the robust objective function to be minimized by the bundle adjustment. This objective function is linearized and conditions for a minimum of the objective are presented. Then, the sparse Levenberg-Marquardt algorithm used to minimize the objective function is presented. Finally, a novel technique for initialization of the globally-referenced bundle adjustment is detailed.

A. Objective Function

The objective function most commonly used in estimation algorithms is the sum of the squares of the measurement residuals, which is commonly referred to as the least-squares objective function. The cost function for a least-squares estimator is defined as

$$\rho_{LS}(r) = \frac{r^2}{2} \tag{22}$$

where r is the normalized measurement residual. This cost function performs well for the CDGPS position measurements because the measurement error distribution is well approximated by a normal distribution.

The image feature measurements, on the other hand, often have large outliers, due to mismatches of the point features and moving point features in the images, and thus the measurement error distribution has much larger tails than a normal distribution. This necessitates the use of a cost function which is robust to outliers in order to obtain an accurate estimate of the state. Ideally, outliers should be entirely suppressed. This can be accomplished using the Tukey bi-weight cost function [17] given as

$$\rho_T(r) = \begin{cases} \frac{c^2}{6}\left(1 - \left(1 - \left(\frac{r}{c}\right)^2\right)^3\right) & ;\ |r| \le c \\ \frac{c^2}{6} & ;\ |r| > c \end{cases} \tag{23}$$

where c is a constant typically set to 4.6851 in order to obtain 95% asymptotic efficiency on the standard normal distribution. However, the Tukey bi-weight cost function is not convex, which can lead to difficulties with convergence if the initial guess for the state is too far from the global minimum of the objective function. Another option is the Huber cost function [17] which does not completely suppress outliers, but is convex and significantly reduces the effect of outliers. The Huber cost function is given as

$$\rho_H(r) = \begin{cases} \frac{r^2}{2} & ;\ |r| \le k \\ k\left(|r| - \frac{k}{2}\right) & ;\ |r| > k \end{cases} \tag{24}$$

where k is a constant typically set to 1.345 in order to obtain 95% asymptotic efficiency on the standard normal distribution.

Huber proposed in [17] that one could begin with the Huber cost function to obtain an estimate of the state with the outliers somewhat suppressed and then perform a few iterations using the Tukey bi-weight cost function to obtain an estimate of the state with completely suppressed outliers. Once enough keyframes are obtained, the initial guesses for the pose of new keyframes and the positions of new point features become so precise that there is no need to begin with the Huber cost function because the initial guesses are close enough that the non-convexity of the Tukey bi-weight cost function is not an issue. At this point, one can transition to simply starting with the Tukey bi-weight cost function. This will be the approach taken in this paper.

In summary, the estimation procedure will first employ the objective function given as

$$f_1\left(\mathbf{X}_{BA}\right) = \sum_{i=1}^{N}\left[\rho_{LS}\left(\left|\left|\Delta\mathbf{x}_{\mathcal{G}}^{A_i}\right|\right|\right) + \sum_{j=1}^{M}\rho_H\left(\left|\left|\Delta\mathbf{s}_{\mathcal{I}_i}^{p_j}\right|\right|\right)\right] \tag{25}$$

where $\Delta\mathbf{x}_{\mathcal{G}}^{A_i}$ and $\Delta\mathbf{s}_{\mathcal{I}_i}^{p_j}$ are the normalized measurement residuals defined as

$$\Delta\mathbf{x}_{\mathcal{G}}^{A_i} = R_{\mathbf{x}_{\mathcal{G}}^{A_i}}^{-1/2}\left(\tilde{\mathbf{x}}_{\mathcal{G}}^{A_i} - \mathbf{x}_{\mathcal{G}}^{A_i}\right) \tag{26}$$

$$\Delta\mathbf{s}_{\mathcal{I}_i}^{p_j} = R_{\mathbf{s}_{\mathcal{I}_i}^{p_j}}^{-1/2}\left(\tilde{\mathbf{s}}_{\mathcal{I}_i}^{p_j} - \mathbf{s}_{\mathcal{I}_i}^{p_j}\right) \tag{27}$$

where $\tilde{\mathbf{x}}_{\mathcal{G}}^{A_i}$ and $\tilde{\mathbf{s}}_{\mathcal{I}_i}^{p_j}$ are the actual CDGPS position and image feature measurements, respectively, and $R_{\mathbf{x}_{\mathcal{G}}^{A_i}}^{-1/2}$ and $R_{\mathbf{s}_{\mathcal{I}_i}^{p_j}}^{-1/2}$ are the inverse of the Cholesky factorization of the measurement covariance matrices for the CDGPS position and image feature measurements, respectively. After the estimation procedure converges to a solution to the problem $\operatorname{argmin}_{\mathbf{X}_{BA}\in\mathcal{S}} f_1\left(\mathbf{X}_{BA}\right)$ where $\mathcal{S} = \left\{\mathbf{X}_{BA}\in\mathcal{R}^{7N+3M}\;\middle|\;\left|\left|\mathbf{q}_{\mathcal{G}}^{\mathcal{C}_i}\right|\right| = 1\ for\ i = 1,2,\ldots,N\right\}$, the objective function is changed to the following

$$f_2\left(\mathbf{X}_{BA}\right) = \sum_{i=1}^{N}\left[\rho_{LS}\left(\left|\left|\Delta\mathbf{x}_{\mathcal{G}}^{A_i}\right|\right|\right) + \sum_{j=1}^{M}\rho_T\left(\left|\left|\Delta\mathbf{s}_{\mathcal{I}_i}^{p_j}\right|\right|\right)\right] \tag{28}$$

The estimation procedure then determines the solution to the problem $\operatorname{argmin}_{\mathbf{X}_{BA}\in\mathcal{S}} f_2\left(\mathbf{X}_{BA}\right)$ using the solution to $\operatorname{argmin}_{\mathbf{X}_{BA}\in\mathcal{S}} f_1\left(\mathbf{X}_{BA}\right)$ as an initial guess. After enough keyframes are obtained, the estimation procedure will simply skip minimizing the first objective function and go straight to the second.

B. Linearized Objective

The optimization problem defined in the previous section is a non-linear, equality-constrained optimization problem. The equality constraints on the quaternions are enforced by reducing the quaternion to minimal form during the iterative updates, as described in Sec. IV, resulting in an unconstrained, non-linear optimization problem at each iteration. At each iteration of the solution procedure, the objective function is linearized about the current best estimate of the state to obtain a linear, unconstrained optimization problem that will be solved at each iteration. The normalized measurement residuals are linearized using the partial derivatives of the measurement models defined in Sec. IV as follows

$$\begin{aligned}\Delta\mathbf{x}_{\mathcal{G}}^{A_i} \approx& \bar{\Delta}\mathbf{x}_{\mathcal{G}}^{A_i} - H^{A_i}\big|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{X}_{BA}\\ =& \bar{\Delta}\mathbf{x}_{\mathcal{G}}^{A_i} - R_{\mathbf{x}_{\mathcal{G}}^{A_i}}^{-1/2}\left(\left.\frac{\partial\mathbf{h}_x\left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_i},\mathbf{q}_{\mathcal{G}}^{\mathcal{C}_i}\right)}{\partial\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_i}}\right|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_i}\right.\\ &\left. + \left.\frac{\partial\mathbf{h}_x\left(\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_i},\mathbf{q}_{\mathcal{G}}^{\mathcal{C}_i}\right)}{\partial\delta\mathbf{e}_{\mathcal{G}}^{\mathcal{C}_i}}\right|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{e}_{\mathcal{G}}^{\mathcal{C}_i}\right)\end{aligned} \tag{29}$$

$$\begin{aligned}\Delta\mathbf{s}_{\mathcal{I}_i}^{p_j} \approx& \bar{\Delta}\mathbf{s}_{\mathcal{I}_i}^{p_j} - H_{\mathcal{I}_i}^{p_j}\big|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{X}_{BA}\\ =& \bar{\Delta}\mathbf{s}_{\mathcal{I}_i}^{p_j} - R_{\mathbf{s}_{\mathcal{I}_i}^{p_j}}^{-1/2}\left(\left.\frac{\partial\mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}_i}^{p_j}\right)}{\partial\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_i}}\right|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{x}_{\mathcal{G}}^{\mathcal{C}_i}\right.\\ &\left. + \left.\frac{\partial\mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}_i}^{p_j}\right)}{\partial\delta\mathbf{e}_{\mathcal{G}}^{\mathcal{C}_i}}\right|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{e}_{\mathcal{G}}^{\mathcal{C}_i} + \left.\frac{\partial\mathbf{h}_s\left(\mathbf{x}_{\mathcal{C}_i}^{p_j}\right)}{\partial\mathbf{x}_{\mathcal{G}}^{p_j}}\right|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{x}_{\mathcal{G}}^{p_j}\right)\end{aligned} \tag{30}$$

Since the objective function is being linearized, the terms involving the image feature measurements in both objectives can be equivalently written as weighted least-squares terms of the form

$$\frac{1}{2}w_V\left(\left|\left|\bar{\Delta}\mathbf{s}_{\mathcal{I}_i}^{p_j}\right|\right|\right)\left|\left|\bar{\Delta}\mathbf{s}_{\mathcal{I}_i}^{p_j} - H_{\mathcal{I}_i}^{p_j}\big|_{\bar{\mathbf{X}}_{BA}}\delta\mathbf{X}_{BA}\right|\right|^2 \tag{31}$$

where $w_V(\cdot)$ is a weight function for the vision measurements that is defined based on the cost function. For the Huber and Tukey cost functions, the corresponding weight functions are

$$w_H(r) = \begin{cases} 1 & ;\ |r|\le k \\ \frac{k}{|r|} & ;\ |r| > k \end{cases} \tag{32}$$

$$w_T(r) = \begin{cases} \left(1-\left(\frac{r}{c}\right)^2\right)^2 & ;\ |r|\le c \\ 0 & ;\ |r| > c \end{cases} \tag{33}$$

C. Conditions for a Minimum of the Linearized Objective

The first-order necessary conditions for a minimum of the objective function state that the derivative with respect to the state must be zero. This condition results in the following set of linear equations, commonly referred to as the normal equations, for the linearized objective function

$$\sum_{i=1}^{N}\left[\left(H^{A_i}\big|_{\bar{\mathbf{X}}_{BA}}\right)^T H^{A_i}\big|_{\bar{\mathbf{X}}_{BA}} + \sum_{j=1}^{M} w_V\left(||\bar{\Delta}\mathbf{s}_{\mathcal{I}_i}^{p_j}||\right)\left(H_{\mathcal{I}_i}^{p_j}\big|_{\bar{\mathbf{X}}_{BA}}\right)^T H_{\mathcal{I}_i}^{p_j}\big|_{\bar{\mathbf{X}}_{BA}}\right]\delta\mathbf{X}_{BA} = \sum_{i=1}^{N}\left[\left(H^{A_i}\big|_{\bar{\mathbf{X}}_{BA}}\right)^T \bar{\Delta}\mathbf{x}_{\mathcal{G}}^{A_i} + \sum_{j=1}^{M} w_V\left(||\bar{\Delta}\mathbf{s}_{\mathcal{I}_i}^{p_j}||\right)\left(H_{\mathcal{I}_i}^{p_j}\big|_{\bar{\mathbf{X}}_{BA}}\right)^T \bar{\Delta}\mathbf{s}_{\mathcal{I}_i}^{p_j}\right] \tag{34}$$

The terms in the summations in Eq. 34 are extremely sparse and should be treated as such for computational efficiency by ignoring all known zero elements in the summation. The terms corresponding to the CDGPS measurements only have non-zero elements in the rows and columns corresponding to the camera pose for that measurement. The terms corresponding to the image feature measurements only have non-zero elements in the rows and columns corresponding to the camera pose and point feature for that measurement. After performing the summations, the resulting matrix equation is of the form

$$\begin{bmatrix} U & W \\ W^T & V \end{bmatrix}\begin{bmatrix} \delta\mathbf{c} \\ \delta\mathbf{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_c \\ \boldsymbol{\epsilon}_p \end{bmatrix} \tag{35}$$

where $\delta\mathbf{c}$ and $\delta\mathbf{p}$ represent the portion of the state update vector corresponding to the camera poses and the point feature locations respectively. The matrices U and V in Eq. 35 are both block diagonal with U having six by six blocks corresponding to each camera pose and V having three by three blocks corresponding to each point feature location. The matrix W is dense and corresponds to the cross terms between the camera poses and the point features observed from those camera poses.

The matrix coefficient in Eq. 35 is also the second derivative of the objective function with respect to the state. The second-order sufficient conditions for a minimum of the objective function state that the second derivative of the objective function must be positive definite. This second derivative matrix is equal to the sum of positive semi-definite matrices and is thus at least positive semi-definite. So long as the full state is observable, this matrix will be positive definite. Conditions for observability are stated in [12] and will be repeated here without proof. Sufficient conditions for observability of the camera poses are that four non-coplanar point features are observed in three images taken from camera locations that are not collinear. Once the sufficient conditions for observability of the camera poses are satisfied, a necessary and sufficient condition for observability of the remaining point feature locations is that each point feature is observed in at least two images where the lines between each camera location and the point feature location are not the same line. Therefore, the solution to Eq. 35 provides the minimizer for the linearized objective function at each iteration provided that these conditions for observability are satisfied.

D. Sparse Levenberg-Marquardt Algorithm

This optimization problem was solved using the sparse Levenberg-Marquardt (LM) algorithm from Appendix 6 of [18]. This algorithm first inflates the diagonal elements of U and V by a multiplicative factor as follows

$$U_{ij}^* = \begin{cases} (1+\lambda)U_{ii} & ;\ i=j \\ U_{ij} & ;\ \text{otherwise} \end{cases} \qquad V_{ij}^* = \begin{cases} (1+\lambda)V_{ii} & ;\ i=j \\ V_{ij} & ;\ \text{otherwise} \end{cases} \tag{36}$$

Upon completion of the iteration, the value of the objective function is checked to see if it decreased. If the objective increased, then λ is multiplied by a certain factor and the iteration is repeated. This has the effect of decreasing the step size and bringing the solution closer to that which would be obtained if each state variable was optimized independently. If the objective decreased, then λ is divided by the same factor and the iteration is accepted. A common value for the update factor for λ is 10.

To solve Eq. 35 after replacing U and V with U^* and V^*, the algorithm exploits the block diagonal structure of V^* in Eq. 35 by pre-multiplying the equation by a special matrix as follows

$$\begin{aligned} &\begin{bmatrix} I & -W\left(V^*\right)^{-1} \\ 0 & I \end{bmatrix}\begin{bmatrix} U^* & W \\ W^T & V^* \end{bmatrix}\begin{bmatrix} \delta\mathbf{c} \\ \delta\mathbf{p} \end{bmatrix} \\ &= \begin{bmatrix} U^* - W\left(V^*\right)^{-1}W^T & 0 \\ W^T & V^* \end{bmatrix}\begin{bmatrix} \delta\mathbf{c} \\ \delta\mathbf{p} \end{bmatrix} \\ &= \begin{bmatrix} I & -W\left(V^*\right)^{-1} \\ 0 & I \end{bmatrix}\begin{bmatrix} \boldsymbol{\epsilon}_c \\ \boldsymbol{\epsilon}_p \end{bmatrix} \end{aligned} \tag{37}$$

where the matrix $\left(V^*\right)^{-1}$ is a block diagonal matrix composed of three by three blocks that are the inverse of the corresponding blocks in V^*. Therefore, the state update vector can be determined as follows

$$\begin{aligned} \delta\mathbf{c} &= \left(U^* - W\left(V^*\right)^{-1}W^T\right)^{-1}\left(\boldsymbol{\epsilon}_c - W\left(V^*\right)^{-1}\boldsymbol{\epsilon}_p\right) \\ \delta\mathbf{p} &= \left(V^*\right)^{-1}\left(\boldsymbol{\epsilon}_p - W^T\delta\mathbf{c}\right) \end{aligned} \tag{38}$$

If implemented properly to exploit the sparsity of $\left(V^*\right)^{-1}$, this algorithm has computational complexity that is linear in the number of point features and cubic in the number of camera poses. With only a few minor changes, one could rewrite this algorithm such that it has computational complexity that is linear in the number of camera poses and cubic in the number of point features. However, the number of point features being tracked at one time tends to number in the thousands for visual SLAM algorithms, while the number of camera poses is typically kept in the tens or hundreds range. This is based on the fact that tracking more point features results in a higher accuracy than retaining more keyframes as shown by Strasdat in [14].

Convergence of the solution is determined by comparing the norm of the state update vector and the change in the objective

at the end of each iteration to threshold values. If both threshold checks are passed, then the algorithm is declared to have converged and the covariance of the state estimate is computed. This covariance is given by the following equation

$$P_{BA} = \begin{bmatrix} P_{cc} & -P_{cc}WV^{-1} \\ -V^{-T}W^{T}P_{cc}^{T} & V^{-T}W^{T}P_{cc}WV^{-1} + V^{-1} \end{bmatrix} \tag{39}$$

where $P_{cc} = \left(U - WV^{-1}W^{T}\right)^{-1}$. Note that the entire covariance matrix is not computed by the bundle adjustment since only the 3×3 blocks on the diagonal of the bottom right sub-matrix are required by the navigation filter, which will be discussed in Sec. VI. Computing the entire covariance matrix would add significantly to the computational requirements of the bundle adjustment. By only computing the necessary elements of the covariance matrix, the covariance computation is comparable to one additional LM iteration.

E. Initialization

A convenient and highly accurate method for initializing an algorithm for combined CDGPS and visual SLAM is to begin with a stand-alone visual SLAM algorithm. A stand-alone visual SLAM algorithm, such as PTAM [11], is capable of determining the camera poses and point feature locations to high accuracy in an arbitrarily-defined local reference frame up to a scale factor. This stand-alone visual SLAM mode can be used on start-up, when CDGPS position measurements are not yet available, and then transitioned into a combined CDGPS and visual SLAM mode.

In [19], Horn derived a closed form solution to the least squares problem of estimating the similarity transformation relating two coordinate systems given estimates of the locations of at least three points in space in both coordinate systems. This transform could be applied directly to the initialization problem at hand using the camera position estimates from the stand-alone visual SLAM algorithm and the CDGPS position measurements if the camera and the GPS antenna were collocated. However, this is obviously physically unrealizable. The distance between the camera and GPS antenna could also easily be accounted for by shifting the camera position estimates from the stand-alone visual SLAM algorithm to the location of the GPS antenna if the coordinate frame for the visual SLAM estimates did not have a scale factor ambiguity, but this is not the case. Therefore, a modified form of the Horn transform must be derived that accounts for this separation between sensors.

For this modified form of the Horn transform, it is assumed that the vector between the camera and the GPS antenna is known in the $\mathcal{C}$ frame and attitude estimates of the system are available in the vision frame (i.e., the frame defined by the stand-alone visual SLAM algorithm), which is true for stand-alone visual SLAM. Given N stand-alone visual SLAM pose estimates and CDGPS position estimates, the objective function to be minimized is given as

$$\begin{aligned} g = & \frac{1}{2}\sum_{i=1}^{N} \Bigg|\Bigg| \frac{1}{\sqrt{s}} \left(\tilde{\mathbf{x}}_{\mathcal{G}}^{A_i} - \mathbf{x}_{\mathcal{G}}^{\mathcal{V}} - R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) R\left(\tilde{\mathbf{q}}_{\mathcal{V}}^{\mathcal{C}_i}\right) \mathbf{x}_{\mathcal{C}}^{A} \right) \\ & - \sqrt{s} R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) \tilde{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i} \Bigg|\Bigg|^2 \end{aligned} \tag{40}$$

where s, $\mathbf{x}_{\mathcal{G}}^{\mathcal{V}}$, and $\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}$ are the scale-factor, translation, and rotation, respectively, parameterizing the transform from the vision frame to the global coordinate system and $\tilde{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}$ and $\tilde{\mathbf{q}}_{\mathcal{V}}^{\mathcal{C}_i}$ are the position and attitude estimates, respectively, from the stand-alone visual SLAM algorithm. Expressing the error in a symmetric form with respect to the scale factor in Eq. 40 has the effect of weighing the errors in the measurements from both coordinate systems equally and results in a more convenient expression for the resulting solution for the transform [19].

The method for solving this estimation problem follows the same procedure as in [19], but results in some additional complications that prevent a completely analytical solution. First, the measurements are averaged and re-expressed as deviations from these averages as follows

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{G}}^{A}} &= \frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{x}}_{\mathcal{G}}^{A_i} \\ \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{\mathcal{C}}} &= \frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i} \\ \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{A}} &= \frac{1}{N}\sum_{i=1}^{N} R\left(\tilde{\mathbf{q}}_{\mathcal{V}}^{\mathcal{C}_i}\right) \mathbf{x}_{\mathcal{C}}^{A} \end{aligned} \tag{41}$$

$$\begin{aligned} \acute{\mathbf{x}}_{\mathcal{G}}^{A_i} &= \tilde{\mathbf{x}}_{\mathcal{G}}^{A_i} - \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{G}}^{A}} \\ \acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i} &= \tilde{\mathbf{x}}_{\mathcal{G}}^{\mathcal{C}_i} - \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{\mathcal{C}}} \\ \acute{\mathbf{x}}_{\mathcal{V}}^{A_i} &= R\left(\tilde{\mathbf{q}}_{\mathcal{V}}^{\mathcal{C}_i}\right) \mathbf{x}_{\mathcal{C}}^{A} - \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{A}} \end{aligned} \tag{42}$$

Substituting these relations into Eq. 40 for the measurements and noting that the linear terms in $\acute{\mathbf{x}}_{\mathcal{G}}^{A_i}$, $\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}$, and $\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}$ sum to zero results in the equation

$$\begin{aligned} g = & \frac{1}{2}\sum_{i=1}^{N} \Bigg[\Bigg|\Bigg| \frac{1}{\sqrt{s}} \left(\acute{\mathbf{x}}_{\mathcal{G}}^{A_i} - R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) \acute{\mathbf{x}}_{\mathcal{V}}^{A_i} \right) - \sqrt{s} R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) \acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i} \Bigg|\Bigg|^2 \\ & + \Bigg|\Bigg| \frac{1}{\sqrt{s}} \left(\boldsymbol{\mu}_{\mathbf{x}_{\mathcal{G}}^{A}} - \mathbf{x}_{\mathcal{G}}^{\mathcal{V}} - R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{A}} \right) - \sqrt{s} R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{\mathcal{C}}} \Bigg|\Bigg|^2 \Bigg] \end{aligned} \tag{43}$$

The second term in Eq. 43 is the only term that contains the translation, $\mathbf{x}_{\mathcal{G}}^{\mathcal{V}}$, and is a quadratic form. Therefore, this term can be minimized by setting the translation to

$$\mathbf{x}_{\mathcal{G}}^{\mathcal{V}} = \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{G}}^{A}} - R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) \left(s\boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{\mathcal{C}}} + \boldsymbol{\mu}_{\mathbf{x}_{\mathcal{V}}^{A}} \right) \tag{44}$$

However, this relation for the translation depends on the rotation and scale factor, which must still be determined. It is interesting to note that the translation is, quite sensibly, just the difference in the means of the positions of the GPS antenna in both coordinate systems. This is the same result as the original Horn transform from [19] with the addition of the term corresponding to the GPS antenna location relative to the camera.

After substituting this solution for the translation into Eq. 43, only the first term remains. This term is then expanded and the scale factor is pulled out of the summations resulting in the following

$$g = \frac{1}{2}\left(\frac{1}{s}S_{\mathcal{G}} - 2D + sS_{\mathcal{V}}\right) \tag{45}$$

where

$$S_{\mathcal{G}} = \sum_{i=1}^{N}\left|\left|\acute{\mathbf{x}}_{\mathcal{G}}^{A_i} - R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right)\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}\right|\right|^2 \tag{46}$$

$$S_{\mathcal{V}} = \sum_{i=1}^{N}\left|\left|\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}\right|\right|^2 \tag{47}$$

$$D = \sum_{i=1}^{N}\left(\left(\acute{\mathbf{x}}_{\mathcal{G}}^{A_i}\right)^T R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right) - \left(\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}\right)^T\right)\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i} \tag{48}$$

The scale factor can be determined by completing the square and setting that term equal to zero. This results in the following relation for the scale factor

$$s = \frac{\sqrt{S_{\mathcal{G}}}}{\sqrt{S_{\mathcal{V}}}} = \sqrt{\frac{\sum_{i=1}^{N}\left|\left|\acute{\mathbf{x}}_{\mathcal{G}}^{A_i} - R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right)\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}\right|\right|^2}{\sum_{i=1}^{N}\left|\left|\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}\right|\right|^2}} \tag{49}$$

Unlike the original Horn transform [19], this solution for the scale factor depends on the rotation because of the term corresponding to the GPS antenna location relative to the camera. However, the solution for the original Horn transform is otherwise the same. It is interesting to note that the solution for the scale factor is, quite sensibly, simply a ratio of metrics describing the spread of the camera positions in both coordinate systems.

After substituting the solution for the scale factor into the objective function, this leaves an objective function that is only a function of the rotation, which is given by the equation

$$g = \sqrt{S_{\mathcal{V}}S_{\mathcal{G}}} - D \tag{50}$$

To aid in solving for the rotation, a useful relation between rotation matrices and quaternions is employed that is given as follows

$$\begin{aligned}\left(\mathbf{x}_1\right)^T R(\mathbf{q})\mathbf{x}_2 &= \left(\mathbf{q}\otimes\mathbf{x}_2\otimes\mathbf{q}^*\right)^T\mathbf{x}_1 \\ &= \left(\mathbf{q}\otimes\mathbf{x}_2\right)^T\left(\mathbf{x}_1\otimes\mathbf{q}\right) \\ &= \mathbf{q}^T\left\{\mathbf{x}_2\right\}^T\left[\mathbf{x}_1\right]\mathbf{q}\end{aligned} \tag{51}$$

where $[\cdot]$ and $\{\cdot\}$ are the quaternion left and right multiplication matrices, respectively, which, in the case of position vectors, are given as

$$[\mathbf{x}] = \begin{bmatrix} 0 & x_3 & -x_2 & x_1 \\ -x_3 & 0 & x_1 & x_2 \\ x_2 & -x_1 & 0 & x_3 \\ -x_1 & -x_2 & -x_3 & 0 \end{bmatrix} \tag{52}$$

$$\{\mathbf{x}\} = \begin{bmatrix} 0 & -x_3 & x_2 & x_1 \\ x_3 & 0 & -x_1 & x_2 \\ -x_2 & x_1 & 0 & x_3 \\ -x_1 & -x_2 & -x_3 & 0 \end{bmatrix} \tag{53}$$

Substituting this relation into the expanded objective function results in

$$\begin{aligned} g = &\sqrt{S_l}\left(\sum_{i=1}^{N}\left(\left|\left|\acute{\mathbf{x}}_{\mathcal{G}}^{A_i}\right|\right|^2 + \left|\left|\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}\right|\right|^2\right)\right. \\ &\left. -2\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right)^T\left(\sum_{i=1}^{N}\left\{\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}\right\}^T\left[\acute{\mathbf{x}}_{\mathcal{G}}^{A_i}\right]\right)\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right)^{1/2} \\ &- \left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right)^T\left(\sum_{i=1}^{N}\left\{\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}\right\}^T\left[\acute{\mathbf{x}}_{\mathcal{G}}^{A_i}\right]\right)\mathbf{q}_{\mathcal{G}}^{\mathcal{V}} + \sum_{i=1}^{N}\left(\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}\right)^T\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i} \end{aligned} \tag{54}$$

In general, there is no analytic solution to the problem of determining the minimizer of Eq. 54. However, it is safe to assume that the solution will be close to the solution of finding the minimizer for $-\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}\right)^T\left(\sum_{i=1}^{N}\left\{\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}\right\}^T\left[\acute{\mathbf{x}}_{\mathcal{G}}^{A_i}\right]\right)\mathbf{q}_{\mathcal{G}}^{\mathcal{V}}$, which is the result from the original Horn transform from [19]. This is primarily due to $\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}$ being greater than $\acute{\mathbf{x}}_{\mathcal{V}}^{A_i}$ in realistic scenarios because the separation between sensors will be smaller than the distance the whole system moves. The solution to this reduced problem is the eigenvector corresponding to the largest eigenvalue of the matrix $\sum_{i=1}^{N}\left\{\acute{\mathbf{x}}_{\mathcal{V}}^{\mathcal{C}_i}\right\}^T\left[\acute{\mathbf{x}}_{\mathcal{G}}^{A_i}\right]$, which is used as an initial guess for an iterative estimator that uses the Newton-Raphson method to determine the rotation. Although difficult to interpret from Eq. 54, it can easily be seen from the solution to the original Horn transform in [19] that the rotation is, quite sensibly, the direction that best aligns the deviations from the means of the camera positions in both coordinate systems.

Once the estimate for the rotation is determined by the iterative estimator, the scale factor and translation can be computed directly from Eqs. 49 and 44 respectively. This estimate of the similarity transform relating the $\mathcal{V}$ frame to the $\mathcal{G}$ frame can be used to transform the camera poses estimated by the stand-alone visual SLAM algorithm into globally-referenced pose estimates, which are then used as an initial guess for the bundle adjustment.

VI. Navigation Filter

While the bundle adjustment is responsible for maintaining a high-precision, globally-referenced map of the environment, it cannot be relied upon for maintaining an estimate of the current pose of the camera due to real-time constraints. Instead, a navigation filter is employed that maintains an estimate of the current pose of the camera using image feature measurements from non-keyframes and inertial measurements from an IMU for propagation of system dynamics. The navigation filter also leverages the map of the environment maintained by the bundle adjustment so that the image feature measurements can be tied to a global reference. In other words, the navigation filter treats image features as fiduciary markers, but with some known error distribution. The remainder of this section outlines the algorithm used for the navigation filter.

A. Measurement Update Step

During the measurement update step, the filter state is temporarily augmented with the image feature positions corresponding to the features with measurements in the current frame. This temporary augmentation of the state vector allows the estimates of the image feature positions to be treated as a prior with error covariances provided by the bundle adjustment. Taking this approach allows the information from the measurement update to be weighted properly between the filter state and the image feature positions.

This type of state augmentation would normally destroy the ability of the filter to perform in real-time. However, there are several steps that can be taken with little loss in performance to reduce the computational burden of the filter that results in only about twice the computational burden over simply taking the image feature position estimates from bundle adjustment as "truth" (i.e., conditioning on the image feature position estimates from bundle adjustment). These steps are as follows

1) The cross-covariances between image feature position estimates from bundle adjustment can be neglected. If these cross-covariances were retained the filter would be required to invert a dense $3M \times 3M$ matrix at most measurement updates because different features would be seen in different frames. This would make real-time operation impossible. By neglecting cross-covariances, the filter only needs to invert M 3×3 matrices, which can also be stored for use in all frames between bundle adjustment updates. Additionally, these cross-covariances between image feature position estimates are small provided that the system moves a fair amount ($\sim$ 10 m), which is required for good observability anyways. This point is demonstrated by the simulations in Sec. VII.
2) Since the cross-covariances between image feature position estimates from bundle adjustment are being neglected, the normal equations for the measurement update have nearly the same sparse structure as the bundle adjustment normal equations discussed in Sec. V-D. By exploiting the sparse structure of the normal equations, a 15×15 system of linear equations corresponding to the filter state is the largest system of equations that must be solved. Under the scenario where the feature position estimates from bundle adjustment were taken as "truth," the filter would also be required to solve a 15×15 system of equations.
3) The image feature positions are marginalized out of the state after the measurement update and not updated within the measurement update. If the image feature positions were maintained in the state the sparsity of the next measurement update would not be preserved and the filter would need to compute the full covariance matrix for the augmented state. Since the feature position measurements are not maintained in the filter between measurement updates, there is also no need to compute the measurement update for these variables.

However, this convenience of reduced computational burden does not come free. The price paid for this ability to operate the filter in real-time is that the filter covariance estimate is no longer consistent because cross-covariances between the image feature positions were ignored. This effect should be small provided that the system is moved a fair amount ($\sim$ 10 m) and the environment is rich with visually recognizable features. An ad-hoc method, such as fudge factors [20], can be used to slightly inflate the covariance in an attempt to maintain filter consistency. Simulations can be used to tune these fudge factors. Note that, while the filter's estimate may not be consistent, the bundle adjustment is not affected by this issue and maintains a consistent estimate of the map.

Given these considerations, the normal equations for the measurement update step are given as

$$\left(\begin{bmatrix} \bar{P}_F^{-1} & 0 \\ 0 & \bar{P}_p^{-1} \end{bmatrix} + \sum_{i=1}^{M} w_T \left(\left|\left| \bar{\Delta}\mathbf{s}_{\mathcal{I}}^{p_j} \right|\right| \right) \left(H_{\mathcal{I}}^{p_j}|_{\bar{\mathbf{X}}_F, \bar{\mathbf{p}}} \right)^T H_{\mathcal{I}}^{p_j}|_{\bar{\mathbf{X}}_F, \bar{\mathbf{p}}} \right) \begin{bmatrix} \delta\mathbf{X}_F \\ \delta\mathbf{p} \end{bmatrix} = \sum_{i=1}^{M} w_T \left(\left|\left| \bar{\Delta}\mathbf{s}_{\mathcal{I}}^{p_j} \right|\right| \right) \left(H_{\mathcal{I}}^{p_j}|_{\bar{\mathbf{X}}_F, \bar{\mathbf{p}}} \right)^T \bar{\Delta}\mathbf{s}_{\mathcal{I}}^{p_j} \tag{55}$$

where $\bar{P}_F$ and $\bar{P}_p$ are the a priori covariance matrices for the filter state and the image feature positions respectively. To reiterate, P_p is approximated as block diagonal with 3×3 blocks corresponding to each point feature position. After performing the summations, these normal equations are given as

$$\begin{bmatrix} U_F & W_F \\ W_F^T & V_F \end{bmatrix} \begin{bmatrix} \delta\mathbf{X}_F \\ \delta\mathbf{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\epsilon}_F \\ \boldsymbol{\epsilon}_p \end{bmatrix} \tag{56}$$

These equations are of nearly the same form as Eq. 35 with the only difference being that the upper left block corresponding to the filter state is dense. This means that the same sparse transformation applied during the bundle adjustment in Eq. 37 can be applied to the filter measurement update. Therefore, the solution to the normal equations for the filter state update and covariance is given by the equations

$$\delta\mathbf{X}_F = \left(U_F - W_F V_F^{-1} W_F^T \right)^{-1} \left(\boldsymbol{\epsilon}_F - W_F V_F^{-1} \boldsymbol{\epsilon}_p \right) \tag{57}$$

$$P_F = \left(U_F - W_F V_F^{-1} W_F^T \right)^{-1} \tag{58}$$

B. Propagation Step

The propagation step utilizes accelerometer and gyro measurements to aid in the propagation of the filter state. The models for these measurements were defined in Sec. IV-C, but the full state dynamics have yet to be defined. Dynamics equations for the accelerometer and gyro biases were given in Eqs. 18 and 20. The time derivative of the differential quaternion is simply

$$\delta\dot{\mathbf{e}}_{\mathcal{G}}^{\mathcal{C}} = \frac{1}{2}\boldsymbol{\omega}_{\mathcal{G}}^{\mathcal{C}} = \frac{1}{2}\left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)R_{\mathcal{C}}^{\mathcal{B}}\left(\tilde{\boldsymbol{\omega}}_{\mathcal{B}} - \mathbf{b}_{\mathcal{B}}^{\omega}\right) + \boldsymbol{\nu}_1^{\omega}\right) \tag{59}$$

The acceleration is significantly more complicated because the IMU and camera are not collocated, which means that the angular velocity of the system couples into the acceleration when expressed using the accelerometer measurements. From rigid body kinematics, the acceleration of the camera can be derived as

$$\begin{aligned}\mathbf{a}_{\mathcal{G}}^{\mathcal{C}} =& \mathbf{a}_{\mathcal{G}}^{\mathcal{B}} - \boldsymbol{\alpha}_{\mathcal{G}}^{\mathcal{C}} \times \left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{\mathcal{B}}\right) - \boldsymbol{\omega}_{\mathcal{G}}^{\mathcal{C}} \times \left(\boldsymbol{\omega}_{\mathcal{G}}^{\mathcal{C}} \times \left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{\mathcal{B}}\right)\right) \\ & - 2\boldsymbol{\omega}_{\mathcal{G}}^{E} \times \mathbf{v}_{\mathcal{G}}^{\mathcal{C}} - \boldsymbol{\alpha}_{\mathcal{G}}^{E} \times \mathbf{x}_{\mathcal{G}}^{\mathcal{C}} - \boldsymbol{\omega}_{\mathcal{G}}^{E} \times \left(\boldsymbol{\omega}_{\mathcal{G}}^{E} \times \mathbf{x}_{\mathcal{G}}^{\mathcal{C}}\right)\end{aligned} \tag{60}$$

where $\boldsymbol{\omega}_{\mathcal{G}}^{E}$ and $\boldsymbol{\alpha}_{\mathcal{G}}^{E}$ are the angular velocity and acceleration of the Earth, respectively, and $\boldsymbol{\alpha}_{\mathcal{G}}^{\mathcal{C}}$ is the angular acceleration of the camera. This equation can be slightly simplified by recognizing that the angular acceleration of the Earth is negligibly small and the angular acceleration of the camera, while not always small, will not be large over significant time intervals and will roughly average to zero. Removing these terms and substituting the accelerometer and gyro measurement models into Eq. 60 results in the equation

$$\begin{aligned}\mathbf{a}_{\mathcal{G}}^{\mathcal{C}} =& R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)R_{\mathcal{C}}^{\mathcal{B}}\left(\tilde{\mathbf{f}}_{\mathcal{B}} - \mathbf{b}_{\mathcal{B}}^{f}\right) - \frac{G_E}{\left|\left|\mathbf{x}_{\mathcal{G}}^{\mathcal{C}}\right|\right|^3}\mathbf{x}_{\mathcal{G}}^{\mathcal{C}} + \boldsymbol{\nu}_1^{f} \\ & - \boldsymbol{\alpha}_{\mathcal{G}}^{\mathcal{C}} \times \left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{\mathcal{B}}\right) - \left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)R_{\mathcal{C}}^{\mathcal{B}}\left(\tilde{\boldsymbol{\omega}}_{\mathcal{B}} - \mathbf{b}_{\mathcal{B}}^{\omega}\right) + \boldsymbol{\nu}_1^{\omega}\right) \\ & \times \left(\left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)R_{\mathcal{C}}^{\mathcal{B}}\left(\tilde{\boldsymbol{\omega}}_{\mathcal{B}} - \mathbf{b}_{\mathcal{B}}^{\omega}\right) + \boldsymbol{\nu}_1^{\omega}\right) \times \left(R\left(\mathbf{q}_{\mathcal{G}}^{\mathcal{C}}\right)\mathbf{x}_{\mathcal{C}}^{\mathcal{B}}\right)\right) \\ & - 2\boldsymbol{\omega}_{\mathcal{G}}^{E} \times \mathbf{v}_{\mathcal{G}}^{\mathcal{C}} - \boldsymbol{\omega}_{\mathcal{G}}^{E} \times \left(\boldsymbol{\omega}_{\mathcal{G}}^{E} \times \mathbf{x}_{\mathcal{G}}^{\mathcal{C}}\right)\end{aligned} \tag{61}$$

To summarize, the dynamics model for the full state is given by

$$\dot{\mathbf{X}}_F = \begin{bmatrix} \dot{\mathbf{x}}_{\mathcal{G}}^{\mathcal{C}} \\ \dot{\mathbf{v}}_{\mathcal{G}}^{\mathcal{C}} \\ \dot{\mathbf{b}}_{\mathcal{B}}^{f} \\ \delta\dot{\mathbf{e}}_{\mathcal{G}}^{\mathcal{C}} \\ \dot{\mathbf{b}}_{\mathcal{B}}^{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{\mathcal{G}}^{\mathcal{C}} \\ \mathbf{a}_{\mathcal{G}}^{\mathcal{C}} \\ \boldsymbol{\nu}_2^{f} \\ \frac{1}{2}\boldsymbol{\omega}_{\mathcal{G}}^{\mathcal{C}} \\ \boldsymbol{\nu}_2^{\omega} \end{bmatrix} \tag{62}$$

Since the time between IMU updates is small ($\sim$10 ms), Euler integration can be used to determine the state update from time k to k+1 as

$$\delta\bar{\mathbf{X}}_F(k) = \Delta t\ \dot{\mathbf{X}}_F\Big|_{\hat{\mathbf{X}}_F(k),\tilde{u}(k+1)} \tag{63}$$

where Δt is the time interval between time k and k+1 and $\tilde{u}(\cdot)$ is the IMU measurement vector given as

$$\tilde{u}(k) = \begin{bmatrix} \tilde{\mathbf{f}}_{\mathcal{B}}(k) \\ \tilde{\boldsymbol{\omega}}_{\mathcal{B}}(k) \end{bmatrix} \tag{64}$$

To determine the propagated covariance, the state transition matrix and process noise Jacobian matrix must first be determined as

$$F(k) = I + \Delta t\ \frac{\partial\dot{\mathbf{X}}_F}{\partial\mathbf{X}_F}\bigg|_{\hat{\mathbf{X}}_F(k),\tilde{u}(k+1)} \tag{65}$$

$$D(k) = \begin{bmatrix} \frac{\partial\dot{\mathbf{X}}_F}{\partial\boldsymbol{\nu}_1^{f}} & \frac{\partial\dot{\mathbf{X}}_F}{\partial\boldsymbol{\nu}_2^{f}} & \frac{\partial\dot{\mathbf{X}}_F}{\partial\boldsymbol{\nu}_1^{\omega}} & \frac{\partial\dot{\mathbf{X}}_F}{\partial\boldsymbol{\nu}_2^{\omega}} \end{bmatrix}\Bigg|_{\hat{\mathbf{X}}_F(k),\tilde{u}(k+1)} \tag{66}$$

The discrete time process noise covariance matrix is then given as

$$\begin{aligned}Q(k) = & \int_0^{\Delta t} F(k)D(k)\begin{bmatrix} \sigma_{f_1}^2 I & 0 & 0 & 0 \\ 0 & \sigma_{f_2}^2 I & 0 & 0 \\ 0 & 0 & \sigma_{\omega_1}^2 I & 0 \\ 0 & 0 & 0 & \sigma_{\omega_2}^2 I \end{bmatrix} \\ & \cdot D^T(k)F^T(k)d\Delta t\end{aligned} \tag{67}$$

Finally, the propagated covariance matrix is computed using the standard covariance propagation equation for the Kalman filter given as

$$\bar{P}_F(k+1) = F(k)P_F(k)F^T(k) + Q(k) \tag{68}$$

VII. Bundle Adjustment Simulations

The bundle adjustment algorithm detailed in Sec. V was implemented and used in a series of simulations designed to probe the limits of its performance. First, the estimability of the bundle adjustment state was probed under a simplistic simulation scenario designed to demonstrate the behavior of the estimator. Next, a more realistic simulation scenario was created where a user carrying the system began outside a building, walked through a hallway in the building with a single turn, and came out the building on the other side. All simulations assumed a 2 cm standard deviation for the CDGPS position estimates and a 1 pixel standard deviation for the image feature measurements. These are reasonable values for the errors on both types measurements. Additionally, the vector between the camera and GPS antenna was set to $\mathbf{x}_{\mathcal{C}}^{A} = [\ 0.1002 \quad -0.1664 \quad -0.0267\ ]\,m$, which is representative of a real prototype system, to make the simulations more representative of actual performance.

A. Estimability Simulations

This simplistic simulation scenario was designed to demonstrate the estimability of the bundle adjustment state. The scenario involves a single cloud of point features, which are uniformly distributed in a sphere with radius r_p, centered on a point a distance d away from the center of a cloud of camera positions, which are uniformly distributed in a sphere with radius r_c. The camera was pointed toward the center of the cloud of point features with some added random pointing dither.

These simulations were all performed using 200 point features and 25 keyframes. The addition of more point features and keyframes would slightly increase the accuracy of the bundle adjustment's solution (up to some geometric limit), but this is not as interesting as varying the other simulation parameters because there is only marginal benefit for such a simplistic scenario.

After fixing the number of point features and keyframes, there are three simulation parameters which could be varied. These parameters are (1) the scale of the simulation preserving the ratio of $r_p : d : r_c$, (2) the ratio $r_c : d$, and (3) the ratio $r_p : d$. A ratio of $1 : 4$ was used in all simulations for the third parameter, $r_p : d$, because there exists a symmetry between this parameter and $r_c : d$, which was allowed to vary, that makes varying both of them unenlightening. The first set of simulations keep the ratio $r_c : d$ constant at a value $1 : 2$ and vary the scale of the simulation, while the second set of simulations hold the scale constant with $d = 20m$ and vary the motion of the camera represented by the ratio $r_c : d$.

1) Simulations Varying Scale: Simulations were performed for a set of values of d between 0.5 m and 200 m, while preserving the ratio of $r_p : d : r_c$ as $1 : 4 : 2$. Figure 2 shows a scatter plot of the camera pose errors from these simulations as a function of d. The norms of the position errors are shown as blue asterisks with the left axis denoting their value. The attitude errors are shown as red pluses with the right axis denoting their values. Figure 3 shows a scatter plot of the norms of the point feature position errors from these simulations as a function of d.

Above about $d = 20m$, the camera position and attitude errors, from Fig. 2, are roughly invariant to scale and the camera position errors are representative of the 2 cm standard deviation used for each dimension of the CDGPS position estimates. This means that, over this region, the camera position estimates are directly limited in accuracy by the accuracy of the CDGPS position estimates. It is important to note the remarkably small attitude error over this region, which is less than 0.1° in most cases. This demonstrates the power of this technique for precise navigation. Unlike the camera pose errors, the point feature position errors decrease linearly with the scale, even in the region above about $d = 20m$. This is a result of the point feature position estimates being based purely on bearing measurements, which are invariant to scale. Therefore, the point feature position errors must decrease linearly with the scale.

Below about $d = 10m$, the estimability of the problem begins to degrade. The attitude errors grow significantly (to over 1° for $d \leq 1m$) and become highly correlated between frames, as can be seen by how tightly packed the attitude errors are on

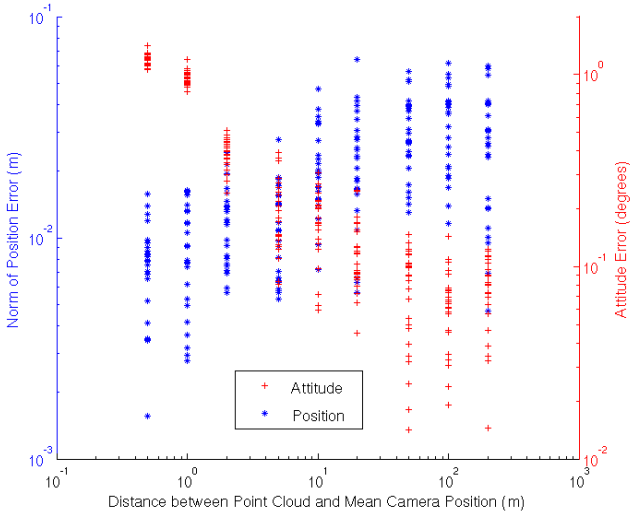

Fig. 2. A scatter plot of the camera pose errors from the simulations as a function of d, while preserving the ratio of $r_p : d : r_c$ as $1 : 4 : 2$. The norms of the position errors are shown as blue asterisks with the left axis denoting their value. The attitude errors are shown as red pluses with the right axis denoting their values.

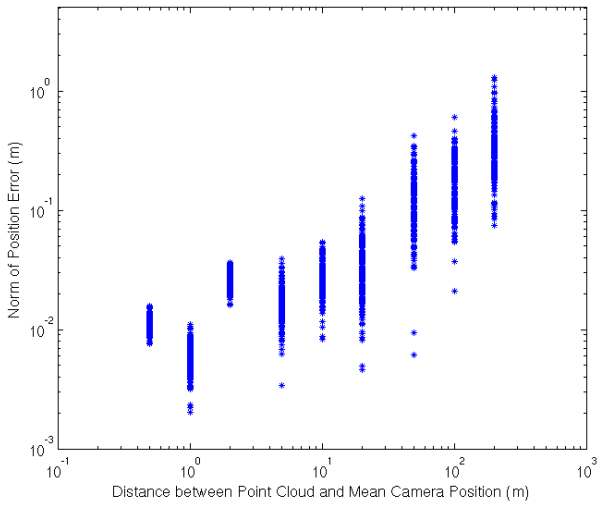

Fig. 3. A scatter plot of the norms of the point feature position errors from the simulations as a function of d, while preserving the ratio of $r_p : d : r_c$ as $1 : 4 : 2$.

the scatter plot. This is because, as the motion of the system decreases, the errors in the CDGPS position estimates start to become significant relative to the distance moved. Therefore, it becomes more difficult to tie the visual SLAM solution to the global coordinate system, resulting in a "bias" between the more accurate local solution normally provided by stand-alone visual SLAM and the global reference frame. This same phenomenon is evident in the point feature position estimates in Fig. 3 for $d = 0.5m$ and $d = 2m$.

A clearer picture of these correlated errors can be seen in the left panel of Fig. 4, which shows the point feature position errors in each dimension for the scenario where $d = 2m$ with the x errors represented by blue asterisks, the y errors

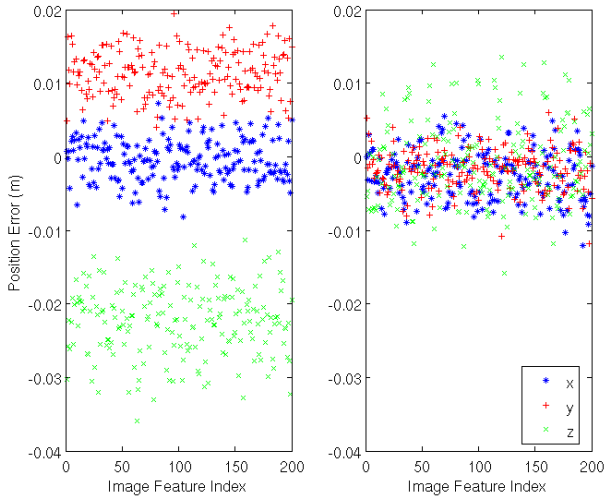

Fig. 4. The left panel shows the point feature position errors in each dimension for the scenario where $d = 2m$ and the ratio of $r_p : d : r_c$ is $1 : 4 : 2$. The x errors are represented by blue asterisks. The y errors are represented by red pluses. The z errors are represented by green x's. The right panel shows the point feature position errors in each dimension under the modified simulation scenario where the camera and point feature positions are spread out over an extra 10 m laterally in one direction.

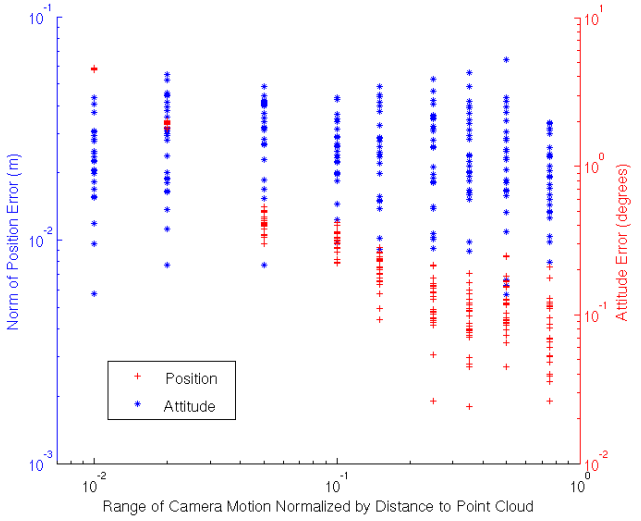

Fig. 5. A scatter plot of the camera pose errors from the simulations as a function of r_c/d with $d = 20m$ and $r_p = 5m$. The norms of the position errors are shown as blue asterisks with the left axis denoting their value. The attitude errors are shown as red pluses with the right axis denoting their values.

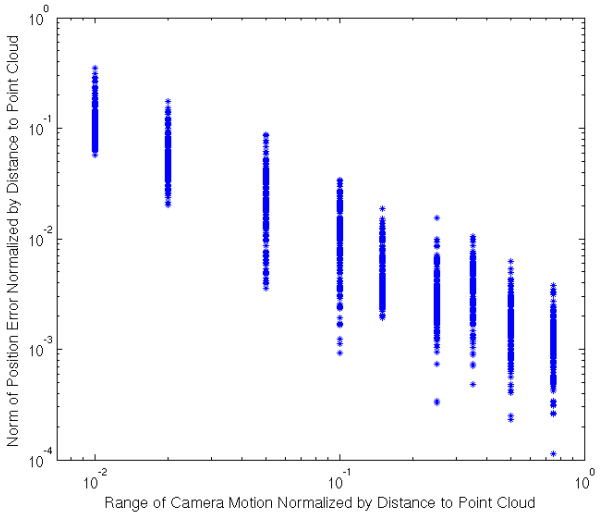

Fig. 6. A scatter plot of the norms of the point feature position errors normalized by d from the simulations as a function of r_c/d with $d = 20m$ and $r_p = 5m$.

represented by red pluses, and the z errors represented by green x's. This correlation is simply an artifact of this simplistic simulation scenario. In reality, one would typically walk by these close-up point features fairly quickly and obtain much better attitude estimates and uncorrelated estimation errors. This fact is partially demonstrated by the right panel of Fig. 4 which displays the point feature position errors under the modified simulation scenario where the camera and point feature positions are spread out over an extra 10 m laterally in one direction. Under this modified scenario, the errors are no longer highly correlated and, although not shown directly, the attitude estimates become much more precise as well.

2) Simulations Varying Camera Motion: Having studied the behavior of the estimator by varying the scale of the problem as a whole, it is informative to now fix the scale and vary the range of motion of the system. The scale of the problem was set by fixing $d = 20m$. Then, the range of motion of the camera was varied from 1% to 75% of d while holding the size of the point cloud constant at $r_p = 5m$. Figure 5 shows a scatter plot of the camera pose errors from these simulations as a function of r_c/d. The norms of the position errors are shown as blue asterisks with the left axis denoting their value. The attitude errors are shown as red pluses with the right axis denoting their values. Figure 6 shows a scatter plot of the norms of the point feature position errors normalized by d from these simulations as a function of r_c/d.

It can be seen from Fig. 5 that below about $r_c/d = 0.2$ the attitude accuracy begins to degrade while positioning accuracy remains roughly constant. This is because the positioning accuracy is directly limited by the accuracy of the CDGPS position estimates for this value of d, as demonstrated in the previous section, while the attitude accuracy is heavily dependent upon the geometry of the problem as a whole. Below about $r_c/d = 0.2$, the estimability of the orientation of the system about roughly the vector connecting the centers of the cloud of point features and cloud of camera positions begins to degrade. This results in an attitude "bias" about this direction which can be seen by how tightly packed the attitude errors are on the scatter plot for small values of r_c/d. It is interesting to note that the pose accuracy, shown in Fig. 5, is invariant to scale above about $d = 20m$, which is the region where the positioning accuracy is directly limited by the accuracy of the CDGPS position estimates.

It can be seen from Fig. 6 that the point feature position errors scale linearly with the decrease in camera motion. Additionally, it is interesting to note that the normalized point feature position errors are invariant over all scales. Therefore, as a rule of thumb, the range of camera motion should be

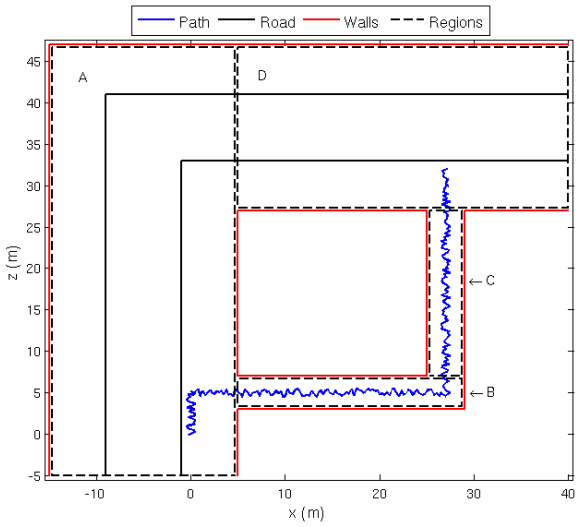

Fig. 7. A diagram showing the layout and camera trajectory for the hallway simulation with the x-axis pointing right, the y-axis pointing into the page, and the z-axis pointing up. The camera trajectory is shown in blue (starting from the bottom left and ending at the top right, while a road and walls of buildings are shown in black and red respectively. Black dashed lines outline four regions of the environment which will be referenced later in the simulation results. These regions are (A) the area the camera is in before entering the hallway, (B) the first leg of the hallway up to and including the turn, (C) the second leg of the hallway, and (D) the area where the camera exits the hallway.

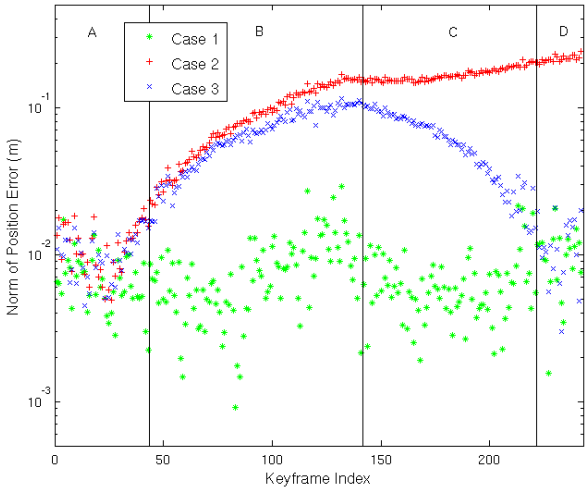

Fig. 8. A plot of the norms of the camera position errors for the hallway simulation for each of the three cases with case 1 represented by green asterisks, case 2 represented by red pluses, and case 3 represented by blue x's. The periods when the camera is in the various regions of the simulation environment are delineated by vertical black lines and labeled.

about 40% of the depth or greater in order to obtain point feature position accuracy to 1% of the depth, as can be seen in Fig. 6. This rule of thumb could be useful in practice when collecting data for the purpose of accurate reconstruction of the environment.

B. Hallway Simulation

Figure 7 shows the layout of this scenario. The camera trajectory, shown in blue in Fig. 7, begins in the bottom left corner of the figure, moves through a hallway, and comes out the other side in the top right. The simulated visual features in this environment included trees lining the road (outlined in black in Fig. 7), the center stripe in the road, windows on the sides of the buildings, entrances and exits to the hallway, doors and posters on the walls inside the hallway, and scattered features on the ceiling of the hallway. In total, there were 1310 features observed in at least 5 keyframes which were included in the bundle adjustment. Keyframes were taken every 0.25 m while moving and every 30° during a turn, resulting in 242 keyframes in total. Uniformly distributed random dither was added to the trajectory in each direction for the position and attitude to simulate human motion. This represents a challenging scenario for the system because there is little lateral motion to improve observability while in the hallway. For convenience, four regions of the simulation environment, shown in Fig. 7 outlined with black dashed lines, are defined to aid in discussion of the results. These regions are (A) the area the camera is in before entering the hallway, (B) the first leg of the hallway up to and including the turn, (C) the second leg of the hallway, and (D) the area where the camera exits the hallway.

Three different cases were run for this scenario regarding the availability of CDGPS position estimates. In the first case, CDGPS position estimates were available at every keyframe. This serves to demonstrate the best performance one could expect from the algorithm in this geometry and acts as a control for the two more interesting cases. The second case assumes that CDGPS position estimates are unavailable for the entire dataset after the camera enters the hallway. This serves to demonstrate the drift one would expect from the system as it moves around indoors. The third and final case assumes that CDGPS position estimates become available again as soon as the system exits the hallway. This serves to demonstrate how the bundle adjustment fixes up the estimates of the poses of previous keyframes and the positions of previously seen point features after GPS is reacquired.

Figures 8 and 9 show the norms of the camera position errors and the camera attitude errors, respectively, for each case with case 1 represented by green asterisks, case 2 represented by red pluses, and case 3 represented by blue x's. The periods when the camera is in the various regions of the simulation environment are delineated by vertical black lines and labeled.

For case 1, the norms of the position errors are mostly under 1 cm and the attitude errors are mostly under 0.1°, which demonstrates the pose accuracy one would expect from this system with open view of the sky. There is one point of interest where the pose errors increase significantly near the end of region B. This point corresponds to the turn inside the hallway when the camera can only see a few point feature, which results in poor estimability of the pose. In fact, there is one keyframe where only 12 point features can be seen. This is partially an artifact of the simulation because, in actual operation, the feature identifier would identify more point features as the camera approached the turn.

In case 2, the attitude estimates are biased by about 0.2° over the entire simulation, as can be seen in Fig. 9, except at

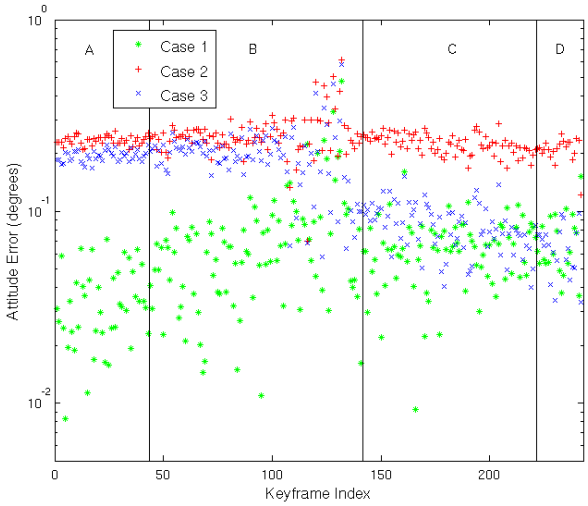

Fig. 9. A plot of the camera attitude errors for the hallway simulation for each of the three cases with case 1 represented by green asterisks, case 2 represented by red pluses, and case 3 represented by blue x's. The periods when the camera is in the various regions of the simulation environment are delineated by vertical black lines and labeled.

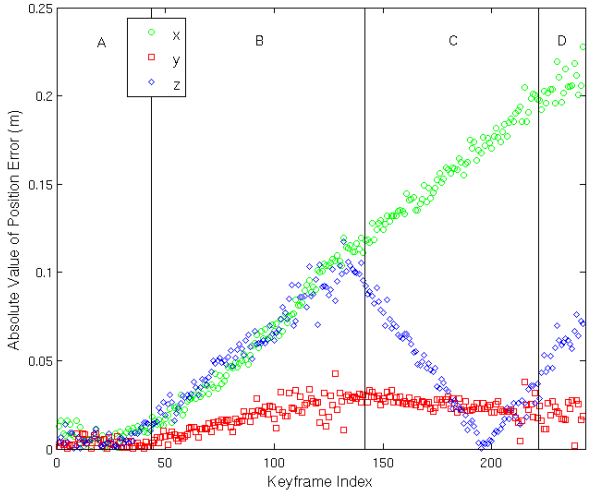

Fig. 10. A plot of the absolute value of the camera position errors in each direction for case 2 of the hallway simulation with the x direction represented by green circles, the y direction represented by red squares, and the z direction represented by blue diamonds. The periods when the camera is in the various regions of the simulation environment are delineated by vertical black lines and labeled.

the turn, where the errors increase substantially. The position errors are close to those from case 1 over most of region A, since these errors are mostly attributed to the CDGPS position estimates which both cases share. However, the position error increases roughly monotonically over the remainder of the simulation, due to the lack of new CDGPS position estimates, reaching a final error of about 0.2 m, which is rather small considering that the camera moved about 50 m after loosing GPS. This is only 0.4% of the distance traveled. There are two terms to this growing error which correspond to an attitude bias, as was noted previously, and a drift error. These two terms can clearly be seen in Fig. 10 which shows the absolute value of the position errors in each direction for this case with the x direction represented by green circles, the y direction represented by red squares, and the z direction represented by blue diamonds. Note how the error in the z direction suddenly begins to decrease just before the end of region B, which is where the turn occurs, and actually goes to zero in the middle of region C before changing signs and increasing again. This type of behavior can clearly be attributed to a bias in attitude. Over the interval the error in the z direction was increasing, the camera traveled about 27 m with an attitude error of about 0.2° resulting in a lateral position error of about 8.7 cm, which is close to the maximum error in the z direction. On the other hand, the errors in the x direction continued to increase after the turn. This error is due to drift of the coordinate system.

For case 3, the camera position errors at the beginning and end of the simulation are comparable to those from case 1 because, as with case 2, the CDGPS position estimates, when present, are the primary limiting factor in the accuracy of the camera position estimate. In the middle of the simulation, when the camera is in the hallway, the position error has a parabolic shape with its maximum just before the end of region B (i.e., at the turn). This demonstrates how the information provided by the CDGPS position estimates is propagated backward by the bundle adjustment to fix up the errors in the camera position estimates. This improvement might have been more drastic if there had been more point features visible during the turn that were also visible in nearby keyframes both before and after the turn. This effectively prevented most of the information gained through the addition of CDGPS position estimates at the end of the simulation from propagated back through the turn. Therefore, little improvement in the camera position estimates from before the turn is achieved over case 2, as can be seen in Fig. 8. This behavior is also observed in the attitude estimates in Fig. 9, which have errors close to those of case 2 before the turn (i.e., near the end of region B) even though there is significant improvement in the attitude errors, over case 2, from after the turn. In fact, the attitude errors from after the turn are of the same order of magnitude as case 1. For comparison with Fig. 10, Fig. 11 shows the absolute value of the position errors in each direction for this case with the x direction represented by green circles, the y direction represented by red squares, and the z direction represented by blue diamonds. Due to the addition of CDGPS position estimates at the end of the data set, the drift of the position estimates while the camera is in the hallway is significantly reduced, as can easily be seen by comparing the errors in the x direction between Figs. 10 and 11.

Figure 12 shows the norms of the point feature position errors for each case divided into subplots based on the region in which the point feature is located and ordered within each region based on the distance of closest approach of the camera to that point feature from closest to furthest. Case 1 is represented by green asterisks, case 2 is represented by red pluses, and case 3 is represented by blue x's. Note that the point feature indices in Fig. 12 are only defined within each region and do not correspond across regions. For region A, cases 2 and 3 have errors that are nearly identical and case 1 only shows significant improvement over the other two cases on some of the point features. The upward trend in the error with feature

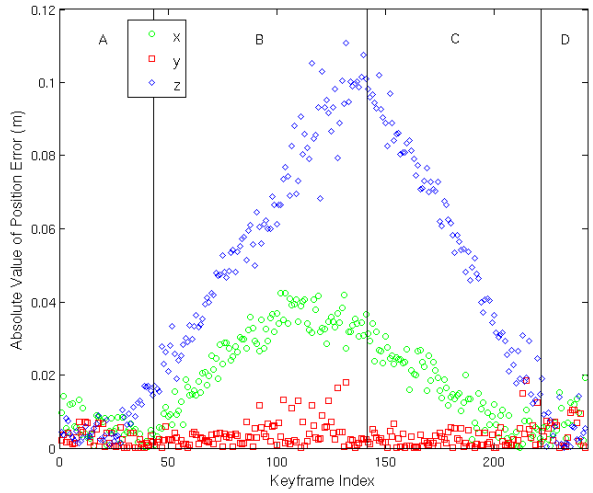

Fig. 11. A plot of the absolute value of the camera position errors in each direction for case 3 of the hallway simulation with the x direction represented by green circles, the y direction represented by red squares, and the z direction represented by blue diamonds. The periods when the camera is in the various regions of the simulation environment are delineated by vertical black lines and labeled.

index is due to the ordering of the features based on the closest approach. For region B, cases 2 and 3 still have nearly identical errors, but case 1 now shows significant improvement over the other cases with about an order of magnitude reduction in error. This is obviously because case 1 is the only case with CDGPS position estimates in this region. The fact that cases 2 and 3 have nearly identical errors in region B demonstrates the fact that the additional information provided by the addition of CDGPS position estimates at the end of the simulation in case 3 was not propagated back through the turn. For region C, there is a clear distinction in accuracy between all three cases with case 1 being the most accurate and case 2 being the least accurate. This shows that the additional information provided by the addition of CDGPS position estimates at the end of the simulation did help case 3 significantly in this region. For region D, cases 1 and 3 have errors that are on the same order, which is to be expected, and case 2 has errors that are significantly larger than the other cases for most of the points features.

VIII. Conclusion

A novel estimation architecture for combined visual SLAM, CDGPS, and inertial navigation was presented that is capable of delivering high accuracy pose ($\sim$ 1 cm positioning accuracy and $\sim 0.1^\circ$ attitude accuracy) and has the potential for real-time operation. The system is centered around a bundle-adjustment-based visual SLAM algorithm that incorporates CDGPS-based position estimates into the bundle adjustment and is responsible for maintaining a highly-accurate, globally-referenced map of the environment. To provide real-time camera pose estimates, a navigation filter is also employed which leverages the map created by bundle adjustment during measurement updates and uses inertial measurements for its propagation step. The system is capable of maintaining highly-accurate, globally-referenced camera pose estimates even without GPS availability for a limited distance of travel.

The globally-referenced bundle adjustment algorithm was implemented and used in a series of simulations. A first set of simulations demonstrated the estimability of the globally-referenced bundle adjustment problem under a simplistic simulation scenario. The second set of simulations represented a fairly realistic scenario where a user carried the system into and through a hallway that blocked reception of the GPS signals. These simulations demonstrated the performance of the bundle adjustment under a challenging scenario. The result was that the bundle adjustment algorithm demonstrated only 0.4% drift in position over 50 m and still attained absolute attitude accuracies of about 0.2°. The addition of CDGPS position estimates at the end of the simulation, once the system had exited the hallway, also resulted in significant improvement in the accuracy of the solution over most of the simulation.

References

[1] J. Rydell and E. Emilsson, "CHAMELEON: Visual-inertial indoor navigation," in *Proceedings of the IEEE/ION PLANS Meeting*. Myrtle Beach, SC: IEEE / Institute of Navigation, April 2012.

[2] G. Nuetzi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular SLAM," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 287–299, Jan. 2011.

[3] A. Soloviev and D. Venable, "Integration of GPS and vision measurements for navigation in GPS challenged environments," in *Proceedings of the IEEE/ION PLANS Meeting*. IEEE/Institute of Navigation, May 2010, pp. 826–833.

[4] D. Zachariah and M. Jansson, "Fusing visual tags and inertial information for indoor navigation," in *Proceedings of the IEEE/ION PLANS Meeting*. Myrtle Beach, SC: IEEE/Institute of Navigation, April 2012.

[5] G. Schall, S. Zollmann, and G. Reitmayr, "Smart vidente: Advances in mobile augmented reality for interactive visualization of underground infrastructure," *Pers Ubiquit Comput*, July 2012.

[6] J. J. Wang, S. Kodagoda, and G. Dissanayake, "Vision aided GPS/INS system for robust land vehicle navigation," in *Proceedings of the ION ITM*. Savannah, GA: Institute of Navigation, Sept. 2009, pp. 600–609.

[7] M. D. Agostino, A. Lingua, D. Marenchino, F. Nex, and M. Piras, "GIMPHI: a new integration approach for early impact assessment," *Applied Geomatics*, vol. 3, no. 4, pp. 241–249, Dec. 2011.

[8] T. Oskiper, S. Samarasekera, and R. Kumar, "Multi-sensor navigation algorithm using monocular camera, IMU and GPS for large scale augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality*. Atlanta, GA: IEEE, Nov. 2012.

[9] J. Wang, M. Garratt, A. Lambert, J. J. Wang, S. Han, and D. Sinclair, "Integration of GPS/INS/vision sensors to navigate unmanned aerial vehicles," *The International Archives of the Photogrammetry, Remote Sensing, and Spatial Information Sciences*, vol. 37, no. B1, pp. 963–969, 2008.

[10] M. Bryson and S. Sukkarieh, "A comparison of feature and pose-based mapping using vision, inertial and GPS on a UAV," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA: IEEE, Sept. 2011.

[11] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2007, pp. 225–234.

[12] D. Shepard, "Fusion of carrier-phase differential GPS, bundle-adjustment-based visual slam, and inertial navigation for precisely and globally-registered augmented reality," Master's thesis, The University of Texas at Austin, May 2013.

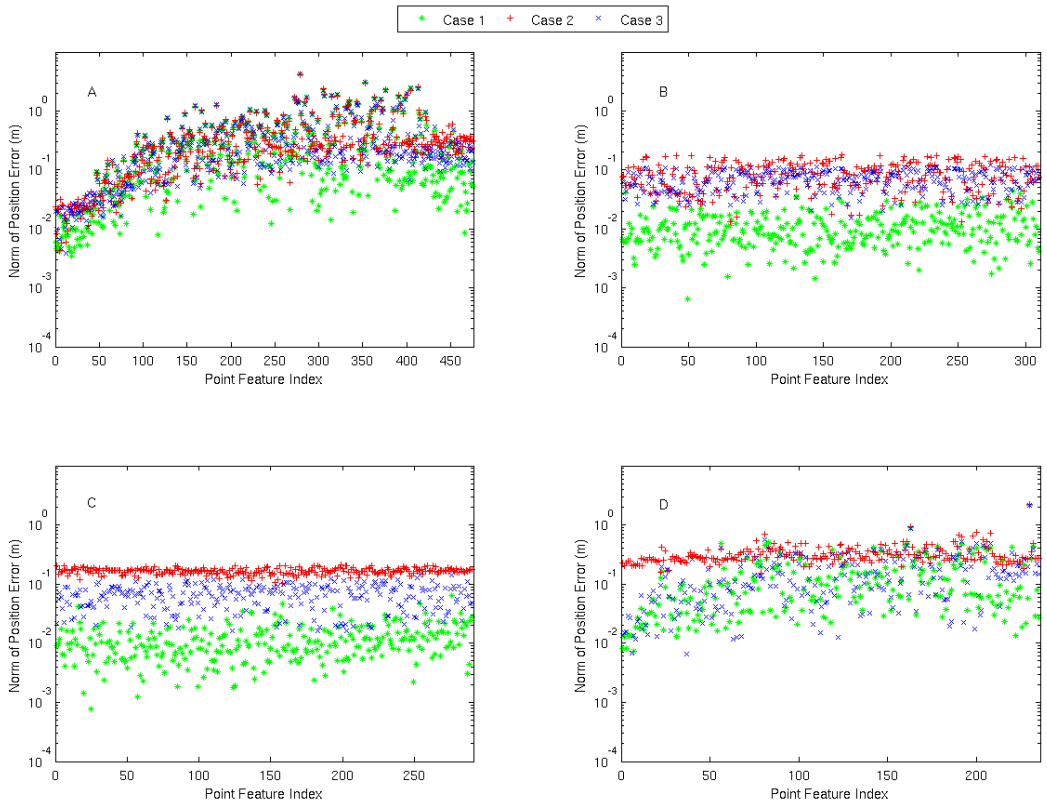

Fig. 12. Plots of the norms of the point feature position errors for each case divided into subplots based on the region in which the point feature is located and ordered within each region based on the distance of closest approach to the point feature from closest to furthest. Case 1 is represented by green asterisks, case 2 is represented by red pluses, and case 3 is represented by blue x's. Note that the point feature indices are only defined within each region and do not correspond across regions.

[13] S. Mohiuddin and M. Psiaki, "Carrier-phase differential Global Positioning System navigation filter for high-altitude spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 801–814, 2008.

[14] H. Strasdat, J. Montiel, and A. J. Davison, "Visual slam: Why filter?" *Image and Vision Computing*, 2012.

[15] T. E. Humphreys, "Attitude determination for small satellites with modest pointing constraints," in *Proc. 2002 AIAA/USU Small Satellite Conference*, Logan, Utah, 2002.

[16] F. Devernay and O. Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, vol. 13, no. 1, pp. 14–24, Aug. 2001.

[17] P. Huber, *Robust Statistics*. New York, New York: John Wiley & Sons, 1981.

[18] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000, vol. 2.

[19] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Wiley and Sons, 2001.