Multi-Sensor Navigation Algorithm Using Monocular Camera, IMU and GPS for Large Scale Augmented Reality

Taragay Oskiper* SRI International
Supun Samarasekera† SRI International
Rakesh Kumar‡ SRI International

Abstract

Camera tracking system for augmented reality applications that can operate both indoors and outdoors is described. The system uses a monocular camera, a MEMS-type inertial measurement unit (IMU) with 3-axis gyroscopes and accelerometers, and GPS unit to accurately and robustly track the camera motion in 6 degrees of freedom (with correct scale) in arbitrary indoor or outdoor scenes. IMU and camera fusion is performed in a tightly coupled manner by an error-state extended Kalman filter (EKF) such that each visually tracked feature contributes as an individual measurement as opposed to the more traditional approaches where camera pose estimates are first extracted by means of feature tracking and then used as measurement updates in a filter framework. Robustness in feature tracking and hence in visual measurement generation is achieved by IMU aided feature matching and a two-point relative pose estimation method, to remove outliers from the raw feature point matches. Landmark matching to contain long-term drift in orientation via on the fly user generated geo-tiepoint mechanism is described.

Index Terms: MEMS IMU, monocular camera, GPS, inertial navigation, sensor fusion, EKF.

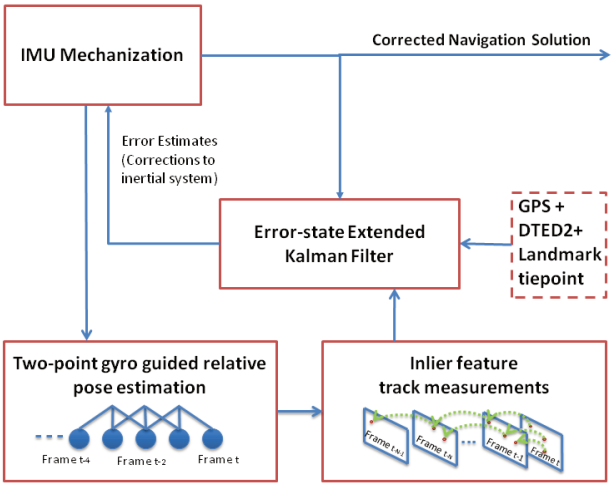

Figure 1: System block diagram.

1 Introduction

In this paper, we present a system using a MEMS-type IMU and GPS unit and a monocular camera for 6 degrees of freedom motion estimation, that can operate both indoors and outdoors, with and without GPS. In contrast to more traditional approaches which first compute a pose estimate from visually tracked features and build a measurement model based on this, we directly work at the feature track level and create pose constraints from each separate track. This circumvents several obstacles that are inherent in the monocular framework. First of all, the scale ambiguity problem in monocular pose estimation does not come into play. Also, uncertainty propagation is treated more systematically, since feature matching covariance is much more easily expressed than pose measurement uncertainty. Pose estimation from monocular feature correspondences over several frames is highly non-linear. Typically, pose covariance estimate is obtained via back propagation of covariance method, where the goal is to deduce the uncertainty in the pose estimate from the covariance of the feature correspondences. In this method, measurement uncertainty tends to be severely underestimated, and outlier rejection becomes very problematic as a result of this poor uncertainty model, since, in order to reject bad pose measurements, one needs a reliable mechanism to compare the predicted pose against the measurement.

Our approach is inspired by that of [6], but it differs in several important ways. We use a helmet mounted lightweight system with a MEMs type IMU and GPS unit from XSens (http://www.xsens.com/en/general/mti-g), which is less expensive and has poorer performance specifications than the Inertial Science ISIS IMU that was used in [6]. Since our IMU has higher drift rates, the filter requires measurement updates more often, precluding us to maintain delayed filtering with multiple camera poses augmented in the state vector. So, unlike [6], we perform a measurement update at every frame and express the measurements in terms of only the previous and current states, hence requiring cloning of only the previous state. This reduces the state vector dimension maintained in the filter, and as demonstrated in our experimental results, its accuracy is quite sufficient and computational cost is significantly reduced, making it possible for real time implementation. Also, with higher drift rates in the IMU, the rapid increase in uncertainty even after short periods of visual tracking outage due to lack of texture or dark scenery (during which navigation relies purely on the IMU) would cause to relax the Mahalanobis distance gating mechanism that is used in [6] for outlier rejection. This leaves the system vulnerable to erroneously accepting outlier feature matches at the end of the outage. Moreover, the quality of our matches are a lot poorer compared to the SIFT approach employed in [6], as we use Harris corner features and correlation window based matching for stringent real time requirements. Therefore, efficient outlier rejection is a high priority. In order to overcome these difficulties, we employ a robust gyro guided two-point relative pose estimation based method at the front end to weed out the outliers before they enter the filter. It is important to note that, except for outlier rejection, these camera pose estimates are not used further down the pipeline.

Error-state (indirect) form of the Kalman filter [12] is used in order to avoid the dynamic modeling of the complex kinematics associated with each specific sensor platform. In this formulation the motion model is derived from integrating the gyro and accelerometer data from the IMU (as performed in the IMU mechanization processing module, cf. Figure 1.), forming the building block for state prediction, rather than the alternative approach of explicitly

*e-mail: taragay.oskiper@sri.com
†e-mail: supun.samarasekera@sri.com
‡e-mail: rakesh.kumar@sri.com

IEEE International Symposium on Mixed and Augmented Reality 2012
Science and Technology Proceedings
5 - 8 November 2012, Atlanta, Georgia
978-1-4673-4662-7/12/$31.00 ©2012 IEEE

modeling the platform dynamics and using the IMU on the measurement side. Since the IMU is able to follow very accurately the high frequency motion of the platform, the indirect Kalman filter operates on the inertial system error propagation equations which evolve smoothly and are much more adequately represented as linear. A big advantage of this approach is that the navigation solution can be transferred to another mobile platform (carrying equivalent sensors) without any change on the sensor fusion algorithms.

As each new frame is received, features are extracted by use of a Harris corner feature detector. These features are matched to previous frame using normalized correlation of image patches around each such feature. Match success or failure is determined based on mutual agreement criterion. This method does not use any thresholding of correlation scores, rather it decides a match is successful if both features in the current and previous frame pick each other as their best (highest scoring) matches [8]. A feature track table that accommodates several hundred frames is used to record the feature locations and maintain the track information and track lengths. After initial matching step, there are usually a great amount of outliers due to false matches. All features that are tracked for at least three frames are input to robust two-point relative pose estimation algorithm that enforces geometric constraints across three frames. At the end of this step, inlier feature tracks are determined based on the comparison of trifocal Sampson error for each feature against a predetermined threshold. Those tracks that fail this test are terminated and reset as fresh features that were newly detected at the current frame. (These features may become inliers in the future).

At every video frame, entire feature track history with inlier tracks from the current frame extending to the past frames in the sequence is made available to the Kalman filter. A separate measurement equation is created for each feature that is tracked for more than three frames. After this step, the measurement equations for all the tracks are stacked to form the final set of measurement model equations which are a function of both the previous state and the current predicted state, so they are relative (local) in nature as opposed to more typical global measurements which are a lot more straightforward to treat. In order to properly handle such relative measurements in the extended Kalman Filter, stochastic cloning framework [11] is employed. The formal approach to handle relative measurements between the previous and current time instants is to express them in terms of motion estimates between the two states and take into account the cross correlation arising from such a formulation. Stochastic cloning provides the framework to process such relative measurements by augmenting the state vector with two copies of the state estimate, one evolving and one stationary clone. The evolving clone is propagated by the process model (just like conventional Kalman filter framework) whereas the stationary clone is kept static and does not evolve. The relative measurement between the previous and current time instant is then expressed as a function of these two states and a Kalman filter update -modified to incorporate the joint covariance of the two clone states- is performed.

Related Work: An early work related to ours is that of [1], which used a differential GPS receiver, compass, gyros and tilt sensor on a static (non-portable) platform. The real-time system did not have a vision sensor and combined rate gyros with a compass and tilt orientation sensor in a hybrid tracker. Large global drift due to compass errors required sensor recalibration, and local registration errors of 2 degrees were achieved. Another related early work [13], described a system which used vision and gyros in 3DOF orientation tracking framework by extracting approximate 2D feature-motion from the inertial data and using vision feature tracking to correct and refine these estimates in the image domain. More recently [3] described a 3DOF camera tracking system that employs a high precision gyroscope and a vision-based drift compensation algorithm by tracking natural features in the outdoor environment using a template matching technique for landmark detection. The drift in yaw is calculated explicitly whenever a landmark is detected and a correction factor is added to the predicted yaw. There are also a variety of wide area augmented reality systems that rely on known models of the environment and edges [2], [10]. In this work, we make no assumption about the environment aside from a relatively coarse DEM (Digital Elevation Model) map to replace GPS elevation values.

2 Extended Kalman Filter Process Model

In our extended Kalman filter, we denote the ground (global coordinate frame) to IMU pose as $\mathbf{P}_{GI} = [\mathbf{R}_{GI} \;\; \mathbf{T}_{GI}]$ such that a point $\mathbf{X}_G$ expressed in the ground frame can be transferred to the IMU coordinates by $\mathbf{X}_I = \mathbf{R}_{GI}\mathbf{X}_G + \mathbf{T}_{GI}$. Accordingly, $\mathbf{T}_{GI}$ represents the ground origin expressed in the IMU coordinate frame, whereas $\mathbf{T}_{IG} = -\mathbf{R}_{GI}^T\mathbf{T}_{GI}$ is the location of the IMU in the ground coordinate frame. In oder to determine the fixed relation between the IMU and camera coordinate systems, which we refer as the IMU to camera pose, $\mathbf{P}_{IC} = [\mathbf{R}_{IC} \; \mathbf{T}_{IC}]$, we use an extrinsic calibration procedure, as developed in [5]. Accordingly, ground to camera pose is determined by the relation, $\mathbf{P}_{GC} = [\mathbf{R}_{IC}\mathbf{R}_{GI} \;\; \mathbf{R}_{IC}\mathbf{T}_{GI} + \mathbf{T}_{IC}]$.

The total (full) states of the filter consist of the IMU location $\mathbf{T}_{IG}$, the gyroscope bias vector $\mathbf{b}_g$, velocity vector $\mathbf{v}_{IG}$ in global coordinate frame, accelerometer bias vector $\mathbf{b}_a$ and ground to IMU orientation $\mathbf{q}_{GI}$, expressed in terms of the quaternion representation for rotation, such that $\mathbf{R}_{GI} = (|q_0|^2 - \|\vec{\mathbf{q}}\|^2)\mathbf{I}_{3\times3} + 2\vec{\mathbf{q}}\vec{\mathbf{q}}^T - 2q_0[\vec{\mathbf{q}}]_\times$, with $\mathbf{q}_{GI} = [q_0 \;\; \vec{\mathbf{q}}^T]^T$ and $[\vec{\mathbf{q}}]_\times$ denoting the skew-symmetric matrix formed by $\vec{\mathbf{q}}$. For quaternion algebra, we follow the notation and use the frame rotation perspective as described in [4]. Hence, the total (full) state vector is given by

$$\mathbf{s} = [\mathbf{q}_{GI}^T \quad \mathbf{b}_g^T \quad \mathbf{v}_{IG}^T \quad \mathbf{b}_a^T \quad \mathbf{T}_{IG}^T]^T . \tag{1}$$

We use the corresponding system model for the state time evolution

$$\dot{\mathbf{q}}_{GI}(t) = \frac{1}{2}(\mathbf{q}_{GI}(t) \otimes \boldsymbol{\omega}(t)), \quad \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t)$$
$$\dot{\mathbf{v}}_{IG}(t) = \mathbf{a}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad \dot{\mathbf{T}}_{IG}(t) = \mathbf{v}_{IG}(t)$$

where $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ are modeled as white Gaussian noise, and $\mathbf{a}(t)$ is acceleration in global coordinate frame, and $\boldsymbol{\omega}(t)$ is the rotational velocity in IMU coordinate frame. Gyroscope and accelerometer measurements of these two vectors are modeled as:

$$\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{2}$$
$$\mathbf{a}_m(t) = \mathbf{R}_{GI}(t)(\mathbf{a}(t) - \mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t) \tag{3}$$

where $\mathbf{n}_g$ and $\mathbf{n}_a$ are modeled as white Gaussian noise and $\mathbf{g}$ is the gravitational acceleration expressed in the global coordinate frame.

State estimate propagation is obtained by the IMU mechanization equations

$$\dot{\hat{\mathbf{q}}}_{GI}(t) = \frac{1}{2}(\hat{\mathbf{q}}_{GI}(t) \otimes \hat{\boldsymbol{\omega}}(t)) \tag{4}$$
$$\dot{\hat{\mathbf{v}}}_{IG}(t) = \hat{\mathbf{R}}_{GI}^T(t)\hat{\boldsymbol{\alpha}}(t) + \mathbf{g}, \tag{5}$$
$$\dot{\hat{\mathbf{T}}}_{IG}(t) = \hat{\mathbf{v}}_{IG}(t), \qquad \dot{\hat{\mathbf{b}}}_g(t) = 0, \qquad \dot{\hat{\mathbf{b}}}_a(t) = 0 \tag{6}$$

with $\hat{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_g(t)$, and $\hat{\boldsymbol{\alpha}}(t) = \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t)$.

Full-state prediction at every frame instant is performed by solving the above system using fourth-order Runge-Kutta numerical integration method (represented by IMU mechanization module in the system block diagram Figure 1). The 15 dimensional Kalman filter error state consists of

$$\delta\mathbf{s} = [\delta\Theta_{GI}^T \quad \delta\mathbf{b}_g^T \quad \delta\mathbf{v}_{IG}^T \quad \delta\mathbf{b}_a^T \quad \delta\mathbf{T}_{IG}^T]^T \tag{7}$$

according to the following relation between the total state and its inertial estimate

$$\mathbf{q}_{GI} = \hat{\mathbf{q}}_{GI} \otimes \delta\mathbf{q}_{GI}, \text{ with } \delta\mathbf{q}_{GI} \simeq \begin{bmatrix} 1 & \delta\Theta_{GI}^T/2 \end{bmatrix}^T$$
$$\mathbf{b}_g(t) = \hat{\mathbf{b}}_g(t) + \delta\mathbf{b}_g(t),\ \mathbf{b}_a(t) = \hat{\mathbf{b}}_a(t) + \delta\mathbf{b}_a(t)$$
$$\mathbf{v}_{IG}(t) = \hat{\mathbf{v}}_{IG}(t) + \delta\mathbf{v}_{IG}(t),\ \mathbf{T}_{IG}(t) = \hat{\mathbf{T}}_{IG}(t) + \delta\mathbf{T}_{IG}(t)$$

based on which we obtain (after some algebra) the following dynamic process model for the error state:

$$\dot{\delta\mathbf{s}} = \mathbf{F}\delta\mathbf{s} + \mathbf{G}\mathbf{n} \tag{8}$$

where

$$\mathbf{F} = \begin{bmatrix} -[\hat{\omega}]_\times & -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\hat{\mathbf{R}}_{GI}^T[\hat{\alpha}]_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_{GI}^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix},$$

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_{wg} \\ \mathbf{n}_a \\ \mathbf{n}_{wa} \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_{GI}^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}$$

During filter operation, ground to IMU pose $\mathbf{P}_{GI}$ is predicted prior to each update instant by propagating the previous estimate using all the IMU readings between the current and previous video frames via IMU mechanization equations. After each update, estimates of the errors (which form the error-states of the filter) are fed-back to correct the predicted pose before it is propagated to the next update and so on.

3 Two-point Gyro Guided Relative Pose Estimation Method for Inlier Detection

Since the IMU is operated at a much higher rate than the video frame rate (120Hz vs 15Hz in our case) and IMU data is received with negligible delay, every time a new image is received, all the gyro readings between this and the previous frame are already available before feature matching is initiated. According to Fig. 1, we can take advantage of this additional information, by establishing a feedback path from the IMU mechanization module, which provides a very good estimate of the relative rotation, $\mathbf{R}_{C_2C_1}$, between the two camera frames, into the feature tracker. The quality of this estimate is mainly dependent on having good estimates of the gyro biases which are being tracked by the Kalman filter. The relative rotation, $\mathbf{R}_{C_2C_1}$, is used in two ways in our system.

Firstly, it is used to guide the feature matching process where we warp each feature point coordinate on the current image by the rotational homography based on $\mathbf{R}_{C_2C_1}$, which is then used to set the center of the search window for feature matching between the previous and current frames. In this fashion, we essentially eliminate the rotational flow from camera motion. This helps to obtain more correct matches with fewer outliers especially when there is abrupt head motion.

Secondly, $\mathbf{R}_{C_2C_1}$ is used in the following way during the two-point hypothesis generation in the RANSAC loop. Let $\mathbf{E}_{C_1C_2}$ be the essential matrix between the current and previous camera coordinate frames. Then, each true feature match has to satisfy the following epipolar constraint:

$$\mathbf{x}_{C_1}^{k\ T}\mathbf{E}_{C_1C_2}\mathbf{x}_{C_2}^{k} = 0 \text{ where } \mathbf{E}_{C_1C_2} = [\mathbf{T}_{C_2C_1}]_\times \mathbf{R}_{C_2C_1}$$

where each feature point vector $\mathbf{x}^k$ is normalized to have unit norm. By rearranging the above relation into the following form

$$\mathbf{x}_{C_1}^{k\ T}\left[\mathbf{R}_{C_2C_1}\mathbf{x}_{C_2}^{k}\right]_\times \mathbf{T}_{C_2C_1} = 0\,, \tag{9}$$

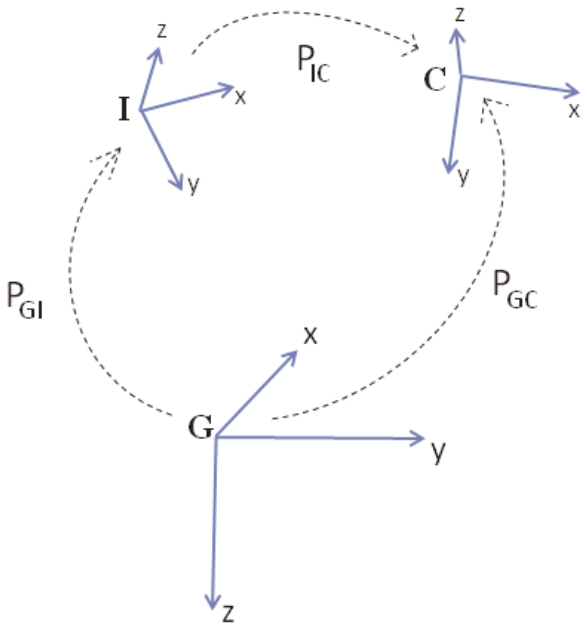

Figure 2: Relation between the global (ground), IMU and camera coordinate frames (denoted by the letters **G**, **I** and **C**, respectively) are shown. Vertical direction, depicted by the z-axis of the ground frame (in the north-east-down, NED, convention), points in the direction of gravity. The origin of the ground coordinate frame coincides with the origin of the camera coordinate frame at start-up, so it is locally defined. IMU and camera are rigidly connected, whose extrinsics are stored in the IMU to camera pose matrix $\mathbf{P}_{IC}$. The system tracks the ground to IMU pose, $\mathbf{P}_{GI}$, which together with $\mathbf{P}_{IC}$, is used in determining ground to camera pose $\mathbf{P}_{GC}$.

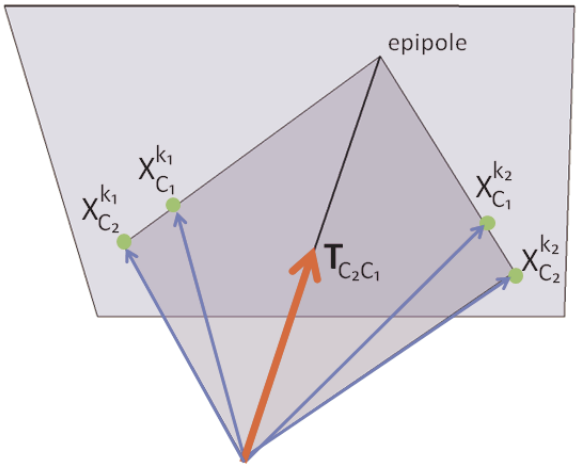

Figure 3: Given the rotation $\mathbf{R}_{C_2C_1}$ between two frames, direction of translation is determined from at least two feature correspondences. This figure illustrates the case when there is no rotation between the two frames, i.e., $\mathbf{R}_{C_2C_1} = \mathbf{I}_{3\times3}$. Each of the two shaded planes in the figure is spanned by two vectors corresponding to each feature correspondence, $\mathbf{x}_{C_1}^{k_i}$ and $\mathbf{x}_{C_2}^{k_i}$ for $i = 1,2$. From (12) the unit norm direction of translation vector is perpendicular to the plane spanned by the cross product $\mathbf{x}_{C_2}^{k_i} \times \mathbf{x}_{C_1}^{k_i}$. This, in turn, forces $\mathbf{T}_{C_2C_1}$ to lie on the intersection of these two planes.

and assuming that the relative rotation $\mathbf{R}_{C_2C_1}$ is known, we write

$$\mathbf{y}^k = -\left[\mathbf{R}_{C_2C_1}\mathbf{x}^k_{C_2}\right]_\times \mathbf{x}^k_{C_1} = -(\mathbf{R}_{C_2C_1}\mathbf{x}^k_{C_2}) \times \mathbf{x}^k_{C_1} \,, \tag{10}$$

and obtain the following relation

$$\mathbf{y}^{kT}\mathbf{T}_{C_2C_1} = 0 \,. \tag{11}$$

The norm, $\|\mathbf{y}^k\|$, of each $\mathbf{y}^k$ is given by the sine of the angle between the two unit norm vectors, $\mathbf{R}_{C_2C_1}\mathbf{x}^k_{C_2}$ and $\mathbf{x}^k_{C_1}$, since it is expressed as the vector cross product of the two. Stacking two $\mathbf{y}^k$ vectors for $k = 1,2$, for which $\|\mathbf{y}^k\|$ exceeds a certain threshold (corresponding to sufficient camera motion condition), and letting $\mathbf{Y} = [\tilde{\mathbf{y}}^1\ \tilde{\mathbf{y}}^2]$, with $\tilde{\mathbf{y}}^k = \mathbf{y}^k/\|\mathbf{y}^k\|$, we get

$$\mathbf{Y}^T\mathbf{T}_{C_2C_1} = 0 \,, \tag{12}$$

so that $\mathbf{T}_{C_2C_1}$ is in the right null-space of the $2x3$ matrix $\mathbf{Y}^T$. Hence, the direction of translation (with sign ambiguity) can be solved as the vector cross product: $\mathbf{T}_{C_2C_1} = \tilde{\mathbf{y}}^1 \times \tilde{\mathbf{y}}^2$.

4 Reprojection Error Based Measurement Model

Each world feature that is observed in more than a single frame introduces a constraint on the associated camera poses. Hence a measurement equation can be formed for each such tracked feature, where the residual is based on its reprojection error on each of the camera's image plane. However, this residual will also depend on the 3D feature position itself which can then be eliminated by combining at least two residuals corresponding to the same feature. In our model, we always use the previous and current camera frames for this purpose. So, assume we have N landmarks, each of which is successfully tracked for at least three frames; current, previous and a reference frame. The reference image frame may not necessarily be common to all the N features, as each track can have its own reference frame that it originated from. If we let $\mathbf{y}^k_G$, $k \in \{1,2,...,N\}$, be the 3D world coordinates of such a landmark with feature coordinates $\mathbf{x}^k_{C_2}$, $\mathbf{x}^k_{C_1}$, and $\mathbf{x}^k_{C_0}$, in the current, previous and reference camera frames, then we have the following relation for the kth tracked feature

$$\mathbf{x}^k_{I_2} = \mathbf{R}_{GI_2}\left(\mathbf{y}^k_G - \mathbf{T}_{I_2G}\right) \tag{13}$$

$$\mathbf{x}^k_{C_2} = \mathbf{R}_{IC}\,\mathbf{x}^k_{I_2} + \mathbf{T}_{IC} \tag{14}$$

where $\mathbf{x}_{I_2}$ stands for the feature location in the current IMU coordinate frame. The 2D feature location measurement on the normalized unit image plane is then expressed as

$$\mathbf{z}^k_2 = h(\mathbf{x}^k_{C_2}) + \nu^k_2 \tag{15}$$

where ν^k_2 is the measurement noise, and h is the perspective projection for the pin-hole camera model, such that given a 3-vector $\mathbf{x} = \begin{bmatrix}\mathbf{x}(1) & \mathbf{x}(2) & \mathbf{x}(3)\end{bmatrix}^T$ in the camera coordinate frame, we have

$$h(\mathbf{x}) = \begin{bmatrix}\mathbf{x}(1)/\mathbf{x}(3) & \mathbf{x}(2)/\mathbf{x}(3)\end{bmatrix}^T \tag{16}$$

Similarly, for the measurement on the previous frame we write

$$\mathbf{z}^k_1 = h(\mathbf{x}^k_{C_1}) + \nu^k_1 \tag{17}$$

Let $\hat{\mathbf{y}}^k_G$ be the estimate of the 3D world coordinate for the kth feature position, such that the total reprojection error on the reference, previous and current frame is minimized, i.e.,

$$\hat{\mathbf{y}}^k_G = \arg\min_{\mathbf{y}^k_G} \sum_{j=0}^{2} \left\|\mathbf{z}^k_j - h(\mathbf{x}^k_{C_j})\right\|^2 \tag{18}$$

where $\mathbf{x}^k_{C_j}$ is defined as (14), for $j \in \{0,1,2\}$ and we use the past pose estimates $\hat{\mathbf{P}}_{GC_0}$ and $\hat{\mathbf{P}}_{GC_1}$ for the reference and previous frame, and the predicted pose $\hat{\mathbf{P}}_{GC_2}$ for the current frame as known constants in the minimization unlike general bundle adjustment. First an initial reconstruction of the 3D position of the feature point is formed by using these pose estimates and triangulating the tracked feature point between the previous and reference frame, and between the current and the reference frame via the efficient triangulation method in [7] and averaging the two estimates. Then, the above least-squares minimization problem can be solved, starting with this initial estimate and performing only two additional Gauss-Newton iterations which we found to be sufficient.

Next, under small error assumption, (13) can be written as

$$\begin{aligned}\mathbf{x}^k_{I_2} \simeq &\left(\mathbf{I}_{3\times3} - [\delta\Theta_{GI_2}]_\times\right)\hat{\mathbf{R}}_{GI_2} \\ &\left(\hat{\mathbf{y}}^k_G + \delta\mathbf{y}^k_G - \hat{\mathbf{T}}_{I_2G} - \delta\mathbf{T}_{I_2G}\right) \,.\end{aligned} \tag{19}$$

Then, using the following

$$\hat{\mathbf{x}}^k_{I_2} = \hat{\mathbf{R}}_{GI_2}\hat{\mathbf{y}}^k_G + \hat{\mathbf{T}}_{GI_2} \text{ and } \delta\mathbf{x}^k_{I_2} = \mathbf{x}^k_{I_2} - \hat{\mathbf{x}}^k_{I_2} \tag{20}$$

and after some manipulation we obtain,

$$\delta\mathbf{x}^k_{I_2} \simeq \hat{\mathbf{R}}_{GI_2}\delta\mathbf{y}^k_G + \left[\hat{\mathbf{x}}^k_{I_2}\right]_\times \delta\Theta_{GI_2} - \hat{\mathbf{R}}_{GI_2}\delta\mathbf{T}_{I_2G} \,. \tag{21}$$

Next, writing

$$\mathbf{x}^k_{C_2} = \mathbf{R}_{IC}\left(\hat{\mathbf{x}}^k_{I_2} + \delta\mathbf{x}^k_{I_2}\right) + \mathbf{T}_{IC} \tag{22}$$

and using the following

$$\hat{\mathbf{x}}^k_{C_2} = \mathbf{R}_{IC}\hat{\mathbf{x}}^k_{I_2} + \mathbf{T}_{IC} \text{ and } \delta\mathbf{x}^k_{C_2} = \mathbf{x}^k_{C_2} - \hat{\mathbf{x}}^k_{C_2} \tag{23}$$

we have,

$$\delta\mathbf{x}^k_{C_2} = \mathbf{R}_{IC}\delta\mathbf{x}^k_{I_2} \,. \tag{24}$$

Finally, from (15) and

$$\hat{\mathbf{z}}^k_2 = h(\hat{\mathbf{x}}^k_{C_2}) \text{ and } \delta\mathbf{z}^k_2 = \mathbf{z}^k_2 - \hat{\mathbf{z}}^k_2 \tag{25}$$

we get

$$\delta\mathbf{z}^k_2 \simeq \mathbf{J}_h(\hat{\mathbf{x}}^k_{C_2})\delta\mathbf{x}^k_{C_2} + \nu^k_2 \,, \tag{26}$$

where for a 3x1 vector $\mathbf{x}$ we have

$$\mathbf{J}_h(\mathbf{x}) = \begin{bmatrix}1/\mathbf{x}(3) & 0 & -\mathbf{x}(1)/(\mathbf{x}(3))^2 \\ 0 & 1/\mathbf{x}(3) & -\mathbf{x}(2)/(\mathbf{x}(3))^2\end{bmatrix} \,. \tag{27}$$

Combining the above with (21) and (24) gives,

$$\delta\mathbf{z}^k_2 \simeq \mathbf{H}^{k,0}_2\delta\mathbf{y}^k_G + \mathbf{H}^{k,1}_2\delta\Theta_{GI_2} + \mathbf{H}^{k,2}_2\delta\mathbf{T}_{I_2G} + \nu^k_2 \tag{28}$$

where we have used $\mathbf{H}^{k,0}_2 = \mathbf{J}_h(\hat{\mathbf{x}}^k_{C_2})\hat{\mathbf{R}}_{GC_2}$, and $\mathbf{H}^{k,1}_2 = \mathbf{J}_h(\hat{\mathbf{x}}^k_{C_2})\mathbf{R}_{IC}\left[\hat{\mathbf{x}}^k_{I_2}\right]_\times$, and $\mathbf{H}^{k,2}_2 = -\mathbf{H}^{k,0}_2$.

In the absence of the first term, $\mathbf{H}^{k,0}_2\delta\mathbf{y}^k_G$, (28) would have been a valid measurement model as a function of the orientation, $\delta\Theta_{GI_2}$, and translation, $\delta\mathbf{T}_{I_2G}$, components of the error-state vector. But this first term can not be ignored, otherwise it would appear that we created a global measurement for error in current orientation and location, out of relative information. Moreover, the fact that the past and current camera pose estimates are used to triangulate the 3D point $\hat{\mathbf{y}}^k_G$ itself, makes it deeply correlated with the filter state, so considering it as part of the noise would violate the assumption that the measurement noise is independent of the filter state.

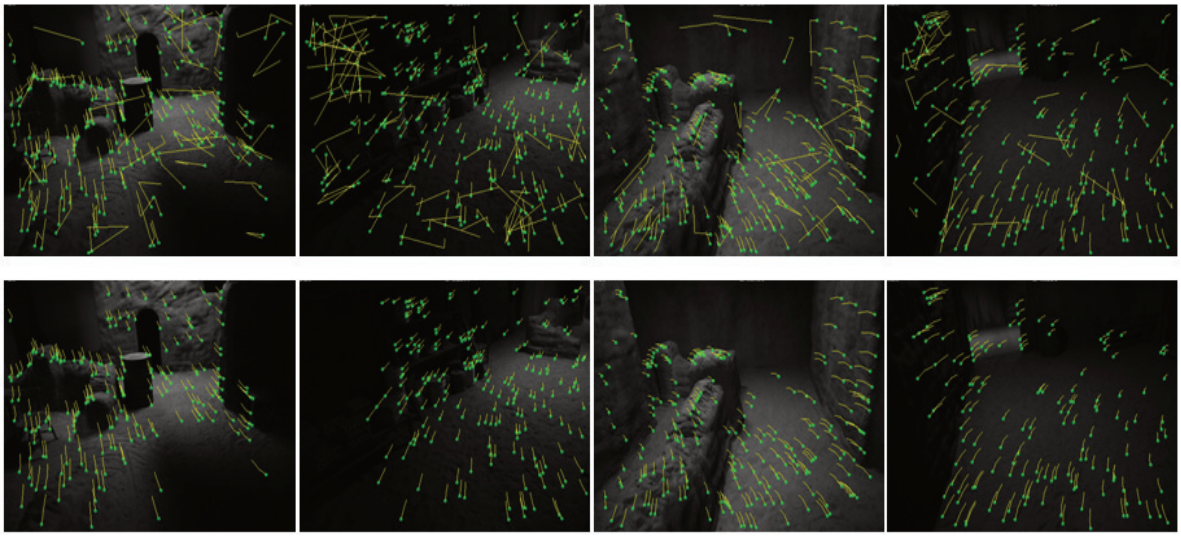

Figure 4: Top images show the raw feature tracks (over three views) obtained in a dataset recorded inside the infantry training facility at Camp Pendleton. Bottom images show the inlier feature tracks for the same frames after the two-point RANSAC based outlier removal. All the inliers enter the Kalman filter as feature track measurements.

Next, we will show how this term can be eliminated to render a valid measurement model by combining the residuals for the same tracked feature over two frames. First, via similar derivation as above, the residual for the k'th feature in the previous frame can be expressed as

$$\delta \mathbf{z}_1^k \simeq \mathbf{H}_1^{k,0} \delta \mathbf{y}_G^k + \mathbf{H}_1^{k,1} \delta \Theta_{GI_1} + \mathbf{H}_1^{k,2} \delta \mathbf{T}_{I_1 G} + \nu_1^k \tag{29}$$

Stacking the residuals of each tracked feature k from the previous and current frame, and letting $\delta \mathbf{z}^k = \begin{bmatrix} \delta \mathbf{z}_1^{k\,T} & \delta \mathbf{z}_2^{k\,T} \end{bmatrix}^T$, and $\nu^k = \begin{bmatrix} \nu_1^{k\,T} & \nu_2^{k\,T} \end{bmatrix}^T$, we get

$$\delta \mathbf{z}^k \simeq \mathbf{H}^{k,0} \delta \mathbf{y}_G^k + \begin{bmatrix} \mathbf{H}_1^k & \mathbf{0}_{2\times 15} \\ \mathbf{0}_{2\times 15} & \mathbf{H}_2^k \end{bmatrix} \begin{bmatrix} \delta \mathbf{s}_1 \\ \delta \mathbf{s}_2 \end{bmatrix} + \nu^k \tag{30}$$

where $\delta \mathbf{s}_1$ corresponds to the Kalman filter error-state vector for the previous frame instant, and $\delta \mathbf{s}_2$ corresponds to the error-state vector for the current frame, and $\mathbf{H}^{k,0} = \begin{bmatrix} \mathbf{H}_1^{k,0\,T} & \mathbf{H}_2^{k,0\,T} \end{bmatrix}^T$, and $\mathbf{H}_1^k = \begin{bmatrix} \mathbf{H}_1^{k,1} & \mathbf{0}_{2\times 9} & \mathbf{H}_1^{k,2} \end{bmatrix}$ and $\mathbf{H}_2^k = \begin{bmatrix} \mathbf{H}_2^{k,1} & \mathbf{0}_{2\times 9} & \mathbf{H}_2^{k,2} \end{bmatrix}$, which follow due to the fact that the error-state vector (7) is 15 dimensional and the non-zero Jacobians act only its orientation and translation components, so the rest of $\mathbf{H}_1^k$ and $\mathbf{H}_2^k$ are set to zero.

If we let $\mathbf{A}$ be a 1x4 left null-vector of the 4x3 matrix $\mathbf{H}^{k,0}$, so that $\mathbf{A}\mathbf{H}^{k,0} = \mathbf{0}$, then by multiplying both sides of (30) with $\mathbf{A}$, we obtain

$$\delta \widetilde{\mathbf{z}}^k \simeq \widetilde{\mathbf{H}}_1^k \delta \mathbf{s}_1 + \widetilde{\mathbf{H}}_2^k \delta \mathbf{s}_2 + \widetilde{\nu}^k \tag{31}$$

where $\delta \widetilde{\mathbf{z}}^k = \mathbf{A}\delta \mathbf{z}^k$, and $\widetilde{\mathbf{H}}_1^k = \mathbf{A} \begin{bmatrix} \mathbf{H}_1^{k\,T} & \mathbf{0}_{2\times 15}^T \end{bmatrix}^T$, and $\widetilde{\mathbf{H}}_2^k = \mathbf{A} \begin{bmatrix} \mathbf{0}_{2\times 15}^T & \mathbf{H}_2^{k\,T} \end{bmatrix}^T$, and $\widetilde{\nu}^k = \mathbf{A}\nu^k$. In this form, (31), establishes a valid Kalman filter measurement equation corresponding to the k'th tracked feature.

The above procedure is repeated for all the true inlier feature tracks for $k \in \{1,2,...,N\}$, and all resulting measurement equations are stacked to form the model:

$$\delta \widetilde{\mathbf{z}} \simeq \widetilde{\mathbf{H}}_1 \delta \mathbf{s}_1 + \widetilde{\mathbf{H}}_2 \delta \mathbf{s}_2 + \widetilde{\nu} \tag{32}$$

where $\delta \widetilde{\mathbf{z}} = \begin{bmatrix} \delta \widetilde{\mathbf{z}}^{1\,T} & \delta \widetilde{\mathbf{z}}^{2\,T} & ... & \delta \widetilde{\mathbf{z}}^{N\,T} \end{bmatrix}^T$, and $\widetilde{\mathbf{H}}_1 = \begin{bmatrix} \widetilde{\mathbf{H}}_1^{1\,T} & \widetilde{\mathbf{H}}_1^{2\,T} & ... & \widetilde{\mathbf{H}}_1^{N\,T} \end{bmatrix}^T$, and similarly for $\widetilde{\mathbf{H}}_2$ and $\widetilde{\nu}$. The final form of the measurement model is given by

$$\delta \widetilde{\mathbf{z}} \simeq \widetilde{\mathbf{H}} \delta \mathbf{s} + \widetilde{\nu} \tag{33}$$

with $\widetilde{\mathbf{H}} = \begin{bmatrix} \widetilde{\mathbf{H}}_1^T & \widetilde{\mathbf{H}}_2^T \end{bmatrix}^T$ and $\delta \mathbf{s} = \begin{bmatrix} \delta \mathbf{s}_1 & \delta \mathbf{s}_2 \end{bmatrix}^T$. This formulation results in a valid Kalman filter measurement model of the error-states, where the current state is augmented with a copy of the previous one to form the final state vector and the filter update equations can be performed using the stochastic cloning approach as in [11].

4.1 Zero-Velocity Constraint

When the camera is standing still, or camera motion is very small between frames, we found the above reprojection error based model insufficient to contain the IMU drift in translation when GPS is not available. In order to handle this situation, we augment the reprojection error model with additional zero velocity constraint based model. The slow to no-motion condition is triggered by monitoring the accelerometer and gyro readings over a few seconds. If the norm of the accelerometer readings are close to the gravitational force, and norm of gyro readings are close to zero during this duration, then the zero velocity model is applied. Since velocity is part of the state vector, zero-velocity measurement model is very straightforward to implement. We directly measure the state, i.e.,

$$\mathbf{z}_v = \mathbf{v}_{IG} + \eta$$

$$\hat{\mathbf{z}}_v = \hat{\mathbf{v}}_{IG}$$

Under no camera motion, we set $\mathbf{z}_v = \mathbf{0}$, hence the residual is given by $-\hat{\mathbf{v}}_{IG}$ and the measurement in the error-state becomes

$$\delta \mathbf{z}_v = \delta \mathbf{v}_{IG} + \eta$$

where η is the measurement noise with covariance Σ_η.

5 GPS AND ELEVATION MEASUREMENT MODEL

Our vision aided inertial navigation algorithm can perform dead-reckoning very effectively via IMU and monocular camera fusion. However, without the absence of global fixes, the drift accumulates over time. In order to overcome this, we have integrated GPS and a global landmark matching mechanism into our solution. In this section, we will review the GPS measurement model that we use when the system is operated outdoors. With the first GPS reading, a reference north-east-down (NED) coordinate system is established and the transform from the earth-centered earth-fixed (ECEF) coordinate frame to this local NED coordinate system, based on the WGS-84 ellipsoid model, is stored in a pose matrix $\mathbf{P}_{ref}$. After this, the initial horizontal position of the IMU is set at the origin of this local frame. The height above ellipsoid (HAE) value on the other hand is obtained from a digital terrain elevation model based lookup and the user entered helmet height above ground. We use the widely available Digital Terrain Elevation Data format level 2 with 30 meter post spacing (DTED2) for this purpose, by converting the terrain elevation above EGM96 geoid as returned by the DTED2 to the one above WGS84 reference ellipsoid. In doing this, we completely ignore the HAE values provided by the GPS unit, since the vertical channel of the GPS is highly erroneous. The terrain elevation model is much more accurate compared to GPS height values but still quite coarse. So they are used with a high covariance in the measurement model.

Having established a Euclidean local world coordinate frame, all subsequent GPS readings are expressed with respect to this coordinate system and used as horizontal position measurements, and height measurements are obtained from the terrain elevation model. With every GPS reading, we first replace the HAE output of the GPS by the value returned from the DTED2 model for that location, and then transform the geodetic position from the GPS latitude and longitude combined with the terrain HAE, into the ECEF coordinate frame. Next, using the reference pose transform matrix $\mathbf{P}_{ref}$, this value is transformed from ECEF coordinate frame into the local reference world frame. After this the measurement model in the error-states becomes very straightforward, since it is a direct function of the location error-state, where the measurement residual is the difference between the GPS measured position and the filter predicted position in the local world frame:

$$\delta \mathbf{z}_t = \delta \mathbf{T}_{IG} + \mathbf{n}$$

where $\mathbf{n}$ is the measurement noise with covariance Σ_n.

6 IMU INITIALIZATION

As mentioned in the previous section, when the system is first turned on, GPS is used to set the initial helmet location at the origin of the locally tangent plane Euclidean reference world coordinate frame that we use for navigation. Global orientation in this coordinate system on the other hand, is obtained by combining the accelerometer and user guided landmarking constraints. The system is provided with the geodetic coordinates of an easily distinguishable 3D landmark that is available in the surrounding environment (we typically use Google earth to pick the landmark coordinates). When landmarking is activated by the user, a cross-hair appears in the center of the video frame and its geodetic coordinates are transformed into the local world coordinate frame using the $\mathbf{P}_{ref}$ coordinate transform matrix as explained in the previous section. The user is required to align this cross-hair with the actual landmark point in the image, by orienting the head toward the landmark direction while keeping the head still and click a mouse button. At this instance, there is very little head motion and the helmet is assumed to be still, hence body acceleration can be ignored. According to (3), the accelerometer readings not only include the body acceleration but also the acceleration due to gravitational force. As for landmark constraints, let $\mathbf{y}$ be the 3D coordinates of the landmark point in the local tangent plane reference coordinate frame and $\mathbf{x}$ be the homogeneous coordinates of the corresponding ray through the image point after radial distortion and intrinsics have been removed. Assuming that the body acceleration is negligible and accelerometer bias is small, which is a valid assumption under the aforementioned conditions, then one can solve for the IMU orientation that can best explain the 3-axis accelerometer readings and the user generated 3D-2D landmark-to-image tiepoint as a constrained least squares problem in the following fashion:

$$\min_{\substack{\mathbf{R}_{GI} \text{ s.t.} \\ \mathbf{R}_{GI}^T \mathbf{R}_{GI} = \mathbf{I}_{3\times 3}}} \|\bar{\mathbf{a}}_m + \mathbf{R}_{GI}\,\bar{\mathbf{g}}\|^2 + \|\bar{\mathbf{x}} - \mathbf{R}_{GI}\,\bar{\mathbf{y}}\|^2$$

where $\bar{\mathbf{g}}$ is the normalized gravity vector in the world coordinate system, $\bar{\mathbf{a}}_m = \mathbf{a}_m / \|\mathbf{a}_m\|$ is the normalized accelerometer readings at the time of the user click event (averaged over few frames), $\bar{\mathbf{x}} = \mathbf{R}_{CI}\,\mathbf{x} / \|\mathbf{R}_{CI}\,\mathbf{x}\|$ is the unit norm image ray expressed in the IMU coordinate frame and $\bar{\mathbf{y}} = \mathbf{y} / \|\mathbf{y}\|$ is the unit norm world point. Using quaternion representation for rotation and the matrix vector product notation for quaternion multiplication [4], together with $\mathbf{R}_{GI}\mathbf{g} = \mathbf{q}_{GI}^* \otimes \mathbf{g} \otimes \mathbf{q}_{GI}$, where $\otimes$ represents quaternion multiplication and $*$ denotes the quaternion conjugate, we can convert the above optimization problem into

$$\min_{\substack{\mathbf{q}_{GI} \text{ s.t.} \\ \|\mathbf{q}_{GI}\|=1}} \left\| \left(\begin{bmatrix} \mathbf{A} \\ \mathbf{X} \end{bmatrix} + \begin{bmatrix} \mathbf{G} \\ -\mathbf{Y} \end{bmatrix} \right) \mathbf{q}_{GI} \right\|^2$$

where

$$\mathbf{G} = \begin{bmatrix} 0 & -g_x & -g_y & -g_z \\ g_x & 0 & -g_z & g_y \\ g_y & g_z & 0 & -g_x \\ g_z & -g_y & g_x & 0 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & -a_x & -a_y & -a_z \\ a_x & 0 & a_z & -a_y \\ a_y & -a_z & 0 & a_x \\ a_z & a_y & -a_x & 0 \end{bmatrix}$$

and where $\mathbf{A}$ and $\mathbf{G}$ are skew-symmetric matrices with $\bar{\mathbf{g}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, $\bar{\mathbf{a}}_m = [a_x\ a_y\ a_z]^T$ and similarly for $\mathbf{X}$ and $\mathbf{Y}$, which can be solved with singular value decomposition and the solution is given as the right singular vector corresponding to the minimum singular value.

7 LANDMARK MATCHING FOR GLOBAL ORIENTATION

We use landmarking in not only determining the initial orientation as described in previous section, but also to increase the precision and reset the orientation drift that may occur in the system. When the user performs the landmark click, the image patch around the clicked point, along with the 3D-2D world point to image correspondence coordinates are stored in a database. Later on, whenever the landmark falls within the field of view of the camera, a search is initiated around the predicted landmark location in the query image. If the search is successful, such that the homography alignment score between the database patch and the query image is high (evaluated by comparing the number of inliers to a threshold), the current location estimate of the landmark in that image is returned by transforming the 2D database point via the computed homography. This active landmark seeking algorithm continuously operates in a background thread and returns the landmark location in the query images which it receives from the main filtering thread. Whenever it is successful, a new 3D to 2D tiepoint is established which can then be used as a measurement in the Kalman filter. However, this tiepoint measurement becomes available with a few frame delay, during which the landmark matching and alignment has been performed. So it has to be propagated to the current image timestamp to be applied in the filter update. There are two options to perform this operation. One approach, which is relatively costly, is to apply the measurement in the correct time in the past and then

propagate the filter output by re-applying all the past measurements (assuming all the past information is buffered). The other simpler approach, which we have chosen, is the following: Instead of propagating the filter output, the measurement itself is propagated to the current timestamp. We transform the 2D location of the landmark point in the past query image to the current image by warping the 2D query image landmark location with the rotational homography determined from the relative rotation estimate between these two image time instances, and thereby create a 3D-2D tiepoint between the landmark and the current image. Since the drift between the two time instances separated by a few frames is negligible in our system, this approach is very effective and simple to implement. Having established a 3D-2D tiepoint, the projective camera measurement model is employed, such that the projection of the 3D landmark point $\mathbf{y}$ onto the normalized image plane is given by

$$\mathbf{z} = h(\mathbf{x}) + \nu \;\text{ with }\; h(\mathbf{x}) = [x_1/x_3 \;\; x_2/x_3]^T \tag{34}$$

where ν is the feature measurement noise with covariance Σ_ν and

$$\mathbf{x} = \mathbf{R}_{GC}\mathbf{y} + \mathbf{T}_{GC} = \mathbf{R}_{IC}\mathbf{R}_{GI}(\mathbf{y} - \mathbf{T}_{IG}) + \mathbf{T}_{IC} \; . \tag{35}$$

Under small error assumption

$$\hat{\mathbf{x}} + \delta\mathbf{x} \simeq \mathbf{R}_{IC}(\mathbf{I} - [\delta\Theta_{GI}]_\times)\hat{\mathbf{R}}_{GI}(\mathbf{y} - \hat{\mathbf{T}}_{IG} - \delta\mathbf{T}_{IG}) + \mathbf{T}_{IC} \; , \tag{36}$$

and

$$\hat{\mathbf{x}} = \mathbf{R}_{IC}\hat{\mathbf{R}}_{GI}(\mathbf{y} - \hat{\mathbf{T}}_{IG}) + \mathbf{T}_{IC} \; . \tag{37}$$

Using (36) and (37), and after some manipulation,

$$\delta\mathbf{x} \simeq \mathbf{R}_{IC}\left[\hat{\mathbf{R}}_{GI}(\mathbf{y} - \hat{\mathbf{T}}_{IG})\right]_\times \delta\Theta_{GI} - \hat{\mathbf{R}}_{GC}\,\delta\mathbf{T}_{IG} \; . \tag{38}$$

Accordingly, if we let $\mathbf{z} = \hat{\mathbf{z}} + \delta\mathbf{z}$, and $\hat{\mathbf{z}} = h(\hat{\mathbf{x}})$, then the measurement equation in the error-states can be written as

$$\delta\mathbf{z} \simeq \mathbf{H}\delta\mathbf{s} + \nu \; , \tag{39}$$

where the measurement Jacobian is given by

$$\mathbf{H} = \mathbf{J}_h[\mathbf{J}_{\Theta_{GI}} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{J}_{\delta\mathbf{T}_{IG}}] \; , \tag{40}$$

with

$$\mathbf{J}_h = \begin{bmatrix} 1/\hat{x}_3 & 0 & -\hat{x}_1/\hat{x}_3^2 \\ 0 & 1/\hat{x}_3 & -\hat{x}_2/\hat{x}_3^2 \end{bmatrix} \; , \tag{41}$$

and

$$\mathbf{J}_{\Theta_{GI}} = \mathbf{R}_{IC}\left[\hat{\mathbf{R}}_{GI}(\mathbf{y} - \hat{\mathbf{T}}_{IG})\right]_\times , \;\text{ and }\; \mathbf{J}_{\delta\mathbf{T}_{IG}} = -\hat{\mathbf{R}}_{GC} \; . \tag{42}$$

8 Experimental Results

We first compare our results against our multi-stereo helmet tracking system that has been rigorously tested over the course of several months as part of the future immersive training environment (FITE) program. In particular, we show results based on the odometry only components of both algorithms on a challenging dataset collected during one of the exercises at IIT Camp Pendleton marine training facility. Figure 5, shows the camera (front-left camera on a two stereo pair system) trajectory obtained on this dataset using landmark matching as described in [9]. Since our system exhibits minimal drift when landmark matching is employed, we treat its output as ground truth for our analysis. The traveled distance is 703 meters and the duration is about 11 minutes long. Feature tracking fails due to dark or textureless regions in about 10% of the frames in the monocular case, and 2% of the frames in stereo. Nevertheless, after both of them are aligned globally to the trajectory obtained by landmark matching, the average deviation between that and monocular trajectory is obtained as 1.16 meters, whereas the result for the stereo algorithm is 1.53 meters. So, in this case, front facing monocular system outperforms the front and back facing stereo method.

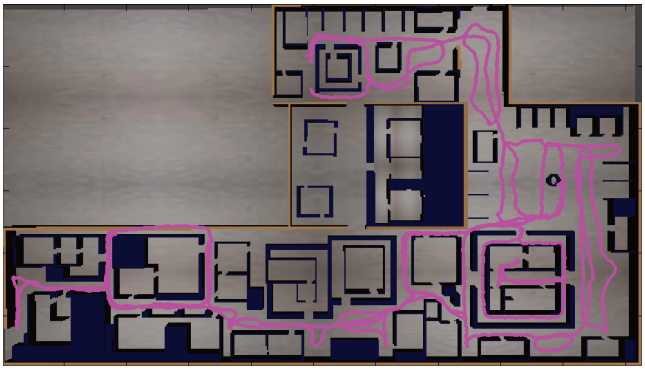

Figure 5: Trajectory obtained with the method in [9] using front and back facing stereo pairs and landmark matching to pre-built landmark database overlaid on the site model.

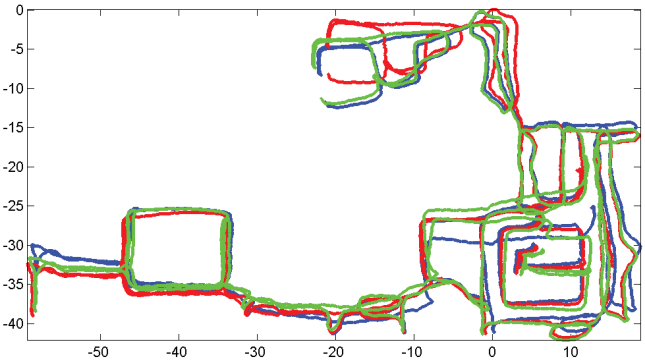

Figure 6: Monocular and stereo trajectories (green and blue respectively) after they have been globally aligned to the trajectory (red) obtained with the algorithm in [9] which uses landmark matching to pre-built database.

Next, we compare our results against the GPS only solution over a long trajectory recorded in SRI Sarnoff campus. This dataset is of 52 minutes long and total traveled distance during this duration is 3607 meters. The user starts his route in the heliport area near the bottom left in Figure 11 and walks around the campus toward the highway intersection at the top and walks back along most of the same path, except near the end, where he enters the building from the south-east entrance and exits from the lounge area on the north-west and completes the route at the same starting position in the heliport area. In Figure 13, we show results from a data collect accompanied by differential gps data in nearby Princeton University campus. Both the trajectories and position errors compared against differential gps data are presented.

Finally, Figures 7 and 8 show the algorithm performance when it is used to estimate camera poses for an augmented reality system that can operate both indoors and outdoors. The first two rows are sample screen shots from datasets without GPS. The bottom two rows, are from datasets where both GPS and geo-landmarking is employed in order to render geo-located air and ground vehicles developed as part of the Augmented Immersive Team Training (AITT) program, (cf. supplementary videos for more examples.) Figure 10 presents quantitative evaluation results in terms of angular insertion errors between rendered and hand labeled marker locations corresponding to the sequence whose sample screen shots are shown in Figure 8.

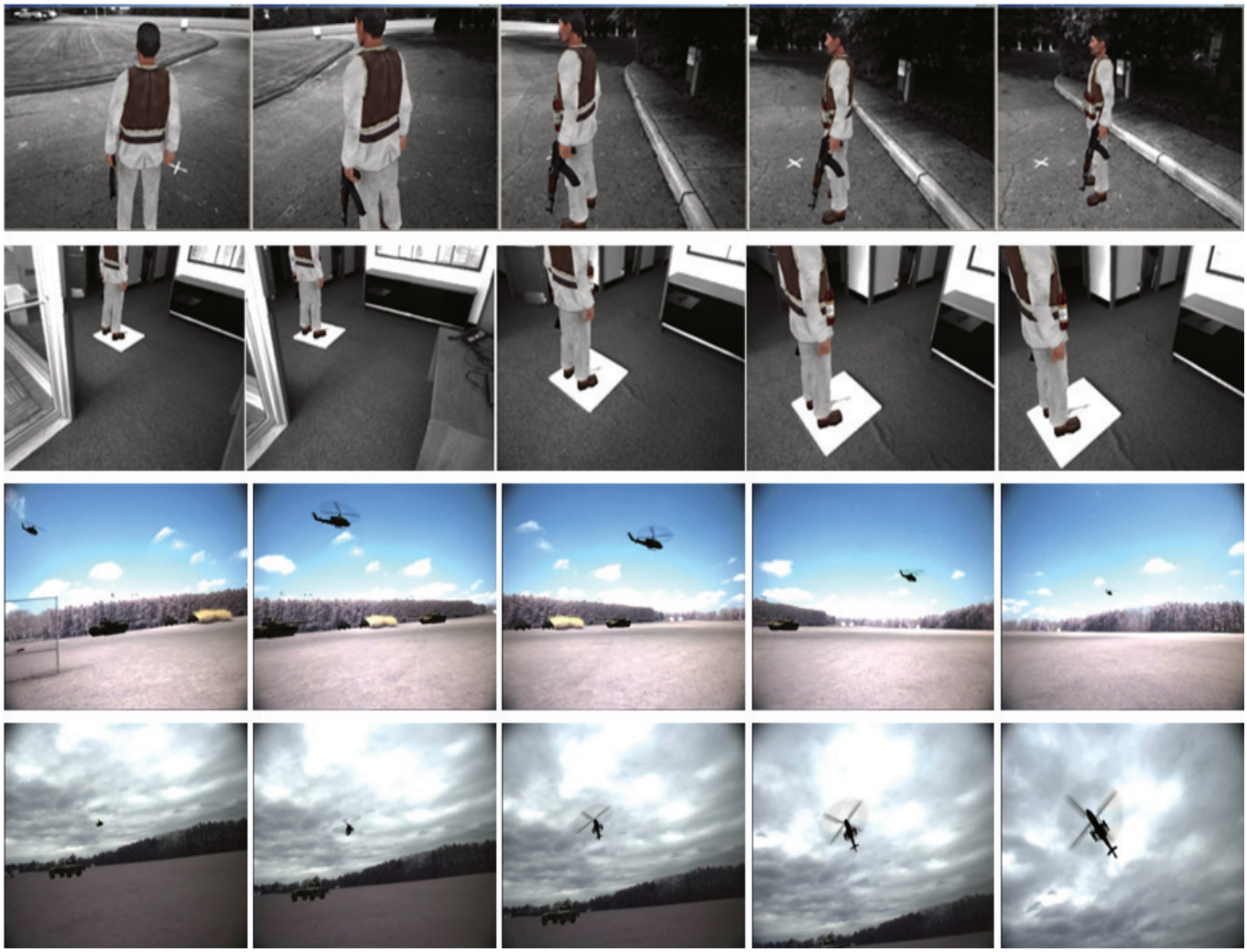

Figure 7: Several augmented reality examples. Top row, outdoors scene in SRI Sarnoff campus, middle row, indoors reception area, bottom row, virtual ground and air vehicles in an outdoors scene in SRI Sarnoff campus (cf. supplementary video).

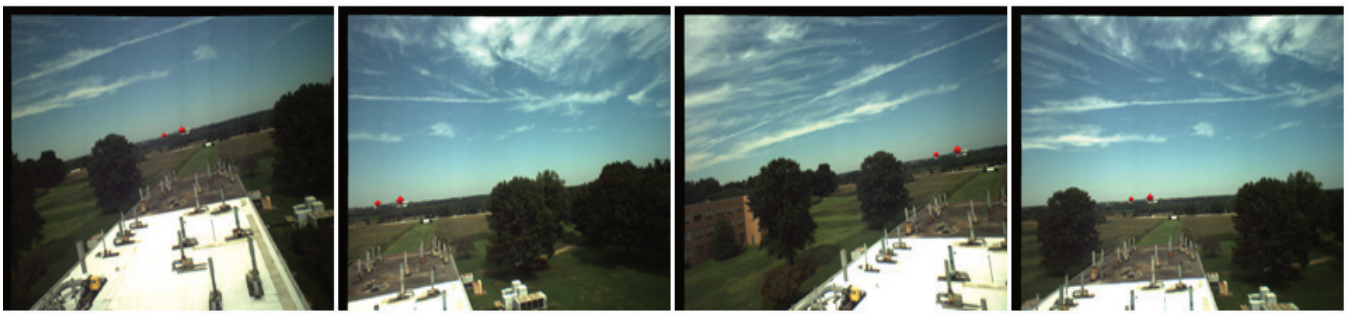

Figure 8: Augmented reality example that is used to measure pose estimation accuracy on a dataset where both GPS and geo-landmarking is employed in order to render geo-located markers representing building corners. The system is operated on the roof of the main Sarnoff building looking toward the Princeton University campus. The red diamonds are the virtual markers overlaid on Fine Hall building and Rockefeller College tower. (cf. supplementary video).

Figure 9: Current system hardware. Sensor package is on the left side and counter balance on the right side of the helmet, with the HMD in front.

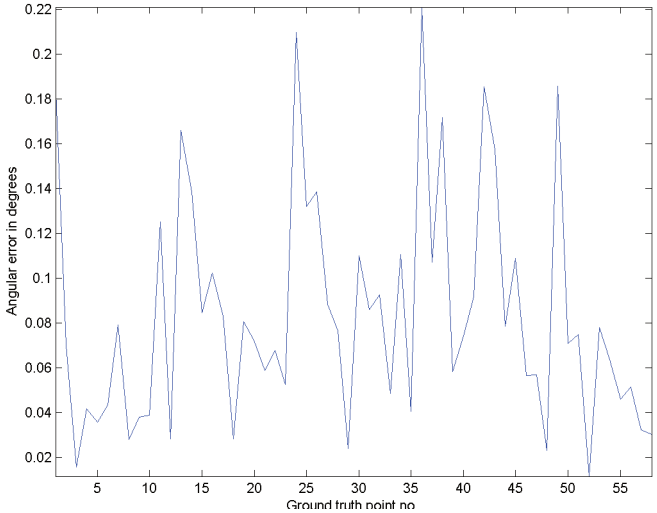

Figure 10: This figure shows a quantitative evaluation of the pose estimation accuracy for an outdoor dataset looking toward Princeton campus from atop Sarnoff building where geo-located targets (with known 3D geodetic coordinates used as inputs to the system) are inserted in the video. Several rendered frames corresponding to this sequence are shown in Figure 8. In order to evaluate the algorithm performance, top corner location of Fine Hall building in Princeton is hand labeled in the video frames at roughly every second (whenever it is in field of view). The plot shows the errors in terms of the angles subtended between the hand labeled expected marker location and the inserted marker location as reported by the system. The video resolution for this dataset is 640x480 pixels. Note, improved accuracies are expected with higher resolution cameras.

9 Conclusion

In this paper we presented a monocular camera tracking system for augmented reality. We demonstrated its accuracy and robustness on a challenging indoor dataset by comparing against a much more powerful system employing front and back facing stereo cameras. Our goal is to achieve same or better performance than that of the stereo platform with a smaller form factor in a GPS denied environment. Toward this end, we are currently working on integrating landmark matching into the monocular tracking algorithm to maintain long term global accuracy and the stability similar to our stereo framework [9]. For outdoors environment, where we can also take advantage of GPS measurements, we demonstrated our algorithm performance by comparing against GPS only solution. For this case, we are also working on a mechanism to dynamically grow the single geo-landmark tiepoint by adding new views to the geo-landmark database, so that we can expand its coverage to handle different viewpoints and scale changes of the same landmark more effectively.

10 Acknowledgments

This material is based upon work supported by ONR Project: Augmented Immersive Team Training (AITT) under Contract N00014-11-C-0433. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ONR. The authors would like to thank Han-Pang Chiu and Xun Zhou for setting up the differential gps system and assistance in the data collections accompanied with differential gps. The authors would also like to thank Richard Schaffer and Sean Cullen of Lockheed Martin, Burlington, MA for the simulation engine used in rendering the synthetic elements for some of the experimental results.

References

[1] R. Azuma, B. Hoff, H. Neely, and R. Sarfaty. A motion-stabilized outdoor augmented reality system. In *IEEE Virtual Reality*, 1999.

[2] B. Jiang, U. Neumann, and S. You. A robust hybrid tracking system for outdoor augmented reality. In *IEEE Virtual Reality*, 2004.

[3] S. K., M. Anabuki, H. Yamamoto, and T. H. A hybrid registration method for outdoor augmented reality. In *IEEE and ACM International Symposium on Augmented Reality*, 2001.

[4] J. B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 1998.

[5] F. M. Mirzaei and S. I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5), 2008.

[6] A. Mourikis and S. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *IEEE ICRA*, 2007.

[7] D. Nister. An efficient solution to the five-point relative pose problem. In *IEEE CVPR*, 2003.

[8] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE conference on Computer Vision and Pattern Recognition*, 2004.

[9] T. Oskiper, H. Chiu, Z. Zhu, S. Samarasekera, and R. Kumar. Stable vision-aided navigation for large-area augmented reality. In *IEEE Virtual Reality Conference*, 2011.

[10] G. Reitmayr and T. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *IEEE ISMAR*, 2006.

[11] S. Roumeliotis and J. Burdick. Stochastic cloning: a generalized framework for processing relative state measurements. In *IEEE ICRA*, 2002.

[12] S. I. Roumeliotis, G. S. Sukhatme, and G. Bekey. Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization. In *IEEE International Conference on Robotics and Automation*, 1999.

[13] Y. S., U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In *IEEE Virtual Reality*, 1999.

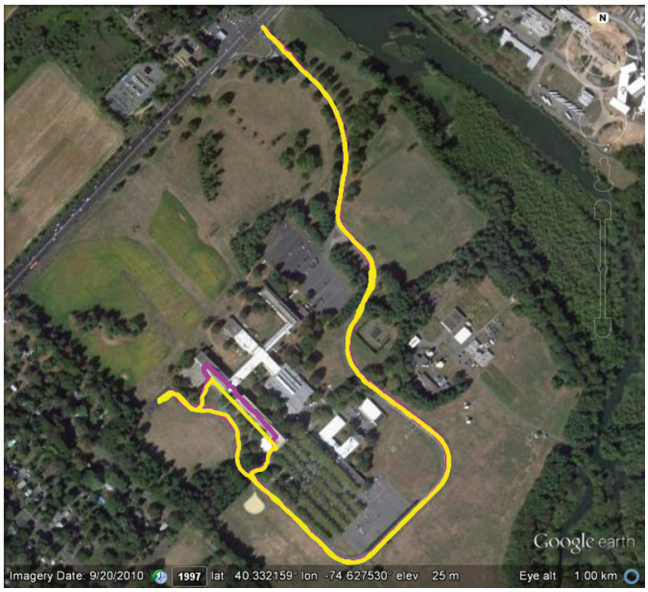

Figure 11: Results on an hour long walk around SRI Sarnoff campus. Magenta shows the system output using IMU, GPS, and monocular camera. Yellow shows the raw GPS based solution. During the portion of the trajectory where the user is inside the building, GPS is not available.

Figure 12: Zoomed in portions of the SRI Sarnoff campus sequence. Top left shows the performance in GPS denied environment where the system exhibits little drift as the user goes through the building. Top right and bottom left portions show that the system output does not get affected by noisy GPS measurements as the user is walking under tree canopy. Bottom right shows a portion where the GPS and filter output are in very close agreement.

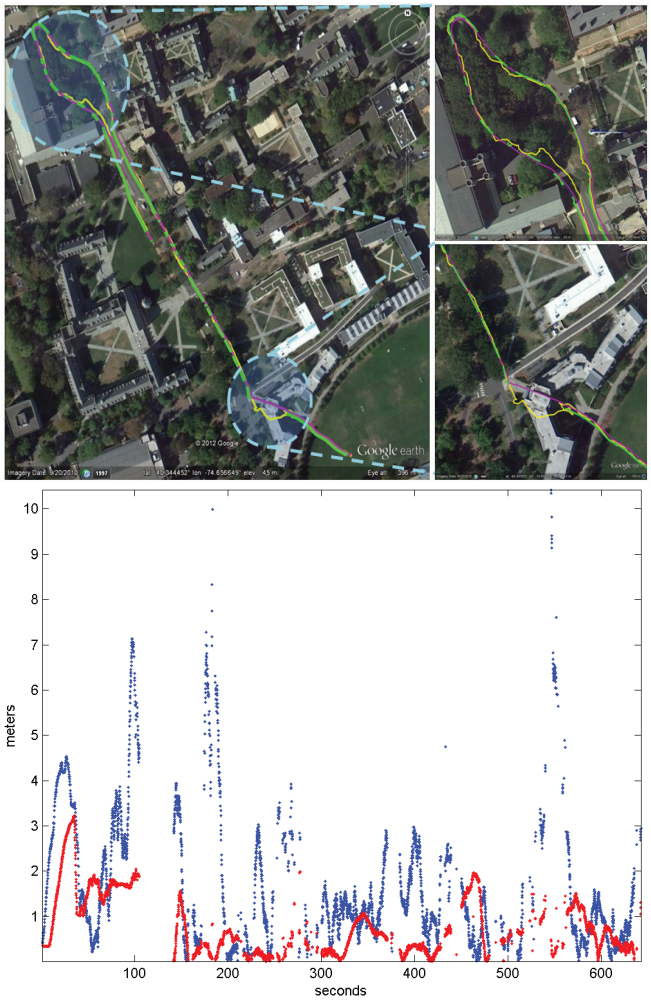

Figure 13: Top figure shows the results from a data collect in nearby Princeton University campus accompanied with differential gps which we treat as ground truth. Magenta trajectory is the output of our system, yellow trajectory is the gps output from XSens MTi-G, and the green trajectory is obtained from differential gps data. Note the gaps in the green path, which correspond to instances when differential gps derived position is not available due to building and tree canopy. On the top left are two zoomed in portions showing detailed views of the highlighted regions. The bottom plot shows the errors in position from our system (in red) and from raw gps data (in blue) computed by comparing the trajectories against differential gps. Note the gaps in the plot where errors can not be calculated due to absence of differential gps data during those instances. However, from the top figure, one can see that, in reality the raw gps errors are much larger in those regions than what is shown in this plot.