Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions

Todd Lupton and Salah Sukkarieh

***Abstract*—In this paper, we present a novel method to fuse observations from an inertial measurement unit (IMU) and visual sensors, such that initial conditions of the inertial integration, including gravity estimation, can be recovered quickly and in a linear manner, thus removing any need for special initialization procedures. The algorithm is implemented using a graphical simultaneous localization and mapping like approach that guarantees constant time output. This paper discusses the technical aspects of the work, including observability and the ability for the system to estimate scale in real time. Results are presented of the system, estimating the platforms position, velocity, and attitude, as well as gravity vector and sensor alignment and calibration on-line in a built environment. This paper discusses the system setup, describing the real-time integration of the IMU data with either stereo or monocular vision data. We focus on human motion for the purposes of emulating high-dynamic motion, as well as to provide a localization system for future human–robot interaction.**

***Index Terms*—Field robots, localization, search-and-rescue robots, sensor fusion.**

I. Introduction

THE motivation of this paper was inspired by the need to develop a human-mounted localization system for first response units, such as fire fighters, counter-terrorism groups, and search-and-rescue operators. The system had to keep track of the location of the operator in an unknown building in real time. The system could not use any purpose built localization infrastructure and had to provide information that could be shared with the future incorporation of robotic systems in the mission.

A human-mounted localization system for first responders operating in built environments poses a number of difficulties. There is the lack of (or poor) global positioning system (GPS) signals, unreliable magnetic readings, high-dynamic motion, and the lack of any internal localization infrastructure. A localization system would, thus, need to use whatever features were available in the environment, but the closeness of objects in the buildings, as well as the presence of walls, means that landmark features may only be observable for short periods of time. The system must also be able to quickly initialize and handle periods when few or no external landmarks are observed.

Cameras can work with landmarks of opportunity that already exist in the environment; therefore, no external infrastructure is required. Indoor Simultaneous Localization and Mapping (SLAM) using a single camera has been developed in the past [4]; however, this implementation had to operate at a high frame rate, i.e., up to 200 Hz [9], and even then, only slow dynamics and certain restricted motions could be used. In addition, as a single camera only provides bearing observations, specialized landmark initialization techniques are required as range is not immediately observable, and the scale of the environment and, therefore, the motions cannot be determined.

The use of a stereo camera pair for SLAM [17], [18] or visual odometry [11] gives very promising results for this kind of application. In addition to providing more constrained motion estimates, the true map scale can be observed and scale drift over the trajectory is eliminated. The close proximity to landmarks and visually rich environments contribute greatly to its success.

The main shortcoming of a system that relies solely on visual observations is that it will fail when sufficient visual landmarks are not observable, even if for just a short period of time. This could easily be the case for human-mounted systems that are considered in this paper. For example, loss of distinct visual features can occur in dark areas of buildings, due to motion blur, if smoke is present, or if the cameras are very close to blank walls such as in narrow corridors or staircases.

The use of an inertial measurement unit (IMU) could help in these momentary periods where visual observations may not be available [19]. Even a low-cost IMU can observe the high dynamics of such a system and can constrain the estimated location for short periods once properly initialized. The difficulty in using an IMU in these applications is one of obtaining proper initialization.

In this paper, the development and analysis of a novel system to process and fuse inertial observations with observations from other body frame sensors, such as cameras, is presented. This system is inspired by the idea that inertial observations can be integrated in an arbitrary frame between poses to form a single pseudo-observation, as presented in [16]. This core idea is expanded in this paper with integration into a graphical filter that

Manuscript received September 7, 2010; revised April 28, 2011; accepted September 21, 2011. Date of publication November 29, 2011; date of current version February 9, 2012. This paper was recommended for publication by Associate Editor J. Neira and Editor D. Fox upon evaluation of the reviewers' comments. This work was supported in part by the Australian Research Council (ARC) Centre of Excellence programme, funded by the ARC and the New South Wales State Government, and by ARC Discovery under Grant DP0665439.

T. Lupton was with The University of Sydney, Sydney, N.S.W. 2006, Australia. He is now with Silverbrook Research, Balmain, N.S.W. 2041, Australia (e-mail: tlup8791@uni.sydney.edu.au).

S. Sukkarieh is with The University of Sydney, Sydney, N.S.W. 2006, Australia (e-mail: salah@acfr.usyd.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2170332

1552-3098/$26.00 © 2011 IEEE

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

allows automatic inertial initialization and map management to produce a robust visual-inertial navigation system.

The main contribution of this paper is that no explicit initialization stage or large uncertainty priors are required, and the initial conditions are automatically recovered in a linear manner. This is achieved by performing the inertial observation mathematical integration in a body fixed frame that moves with the vehicle. Postintegration, both gravity and initial velocity are linearly corrected in the navigation frame, the orientation of which is fixed to the initial vehicle orientation. This way, the gravity vector is estimated directly instead of estimating initial roll and pitch, thus removing the major source of nonlinearity from inertial navigation.

As a consequence, the initial velocity and gravity vector in this frame can be recovered quickly and linearly after the observations have been integrated. These developments lead to a quickly initializing and accurate navigation solution that is tolerant to faults and can quickly recover from errors automatically.

Section II of this paper provides a background on inertial navigation and SLAM techniques and discusses their relevance to the considered application. Section III details the development of the new inertial observation processing algorithm, which we call inertial preintegration, while Section IV explains the calculation of its covariance and Jacobian components that are required for implementation.

In Section V, the concept of performing inertial navigation in a body referenced navigation frame instead of the traditional globally referenced frame is discussed. The reasons why this approach works for the considered application are explained as well as the advantages gained from it. Section VI gives an overview of the proposed inertial navigation technique as a whole discussing how all the components work together to make the initial conditions linearly observable.

Section VII details the implementation of the developed visual-inertial navigation system as a whole. While Section VIII provides a number of experimental results analyzing the performance of the system in real-world environments.

A conclusion is then provided in Section IX-A and future work in Section IX-B.

II. Aided Inertial Navigation

A. Inertial Navigation

IMUs were primarily developed and used for navigation in the aerospace community [21]. In these applications, a globally referenced and fixed navigation frame is used with aiding observations, such as GPS and magnetometer observations, and the navigation solution is computed in this frame.

Acceleration and rotation rate observations are taken from the IMU and transformed into this navigation frame before being integrated to provide an estimate of the current position, velocity, and attitude of the platform. In order for these estimates to be obtained, the initial position, velocity, and attitude of the platform in the frame must first be determined. This is known as the initial condition requirement.

This process is illustrated in Fig. 1, where a number of poses of interest are shown. The dotted line between the poses represents the trajectory of the vehicle and the inertial observations taken at these points. The inertial observations are then transformed into the navigation frame shown by the dashed lines before being integrated to obtain the navigation solution.

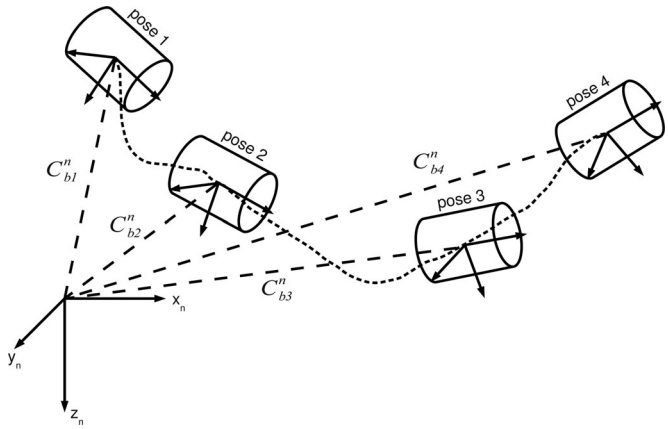

Fig. 1. Inertial navigation in a globally referenced navigation frame. Dotted line represents the trajectory of the vehicle and the inertial observations taken at these points. Inertial observations are transformed into the navigation frame shown by the dashed lines.

The initial attitude estimate is particularly important in order to conduct gravity compensation because of its nonlinear effect on the inertial observation integration. This implies that either an initialization procedure or method of obtaining these initial conditions is required.

These initial conditions can be obtained by an initialization routine that places the vehicle at a known position, velocity, and attitude to use as a starting point [3]. Other options also include collecting IMU and aiding observations, such as GPS, over a period of time and then performing a nonlinear batch initialization to obtain the initial conditions or to just start with a large uncertainty and try to converge to the true state of the vehicle over time [13], [20].

B. Using Inertial Navigation in Simultaneous Localization and Mapping

The algorithms for inertial navigation have been adapted for use in SLAM applications [3], [8], [12], where observations are made in a body referenced frame, such as with camera observations.

These applications are usually only conducted over small areas, where the curvature of the earth can be ignored and low-cost IMUs are used such that earth rotation does not have a significant effect. As a result, the inertial integration that is used for navigation can be simplified; see (1)–(3), shown below, for the position, velocity, and attitude estimates, respectively.

The C^n_{bt} and E^n_{bt} terms are the rotation and rotation rate matrices, respectively, from the current body frame to the navigation frame using the current vehicle orientation ϕ^n_t

$$p^n_{t2} = p^n_{t1} + (t2 - t1)v^n_{t1} + \iint_{t1}^{t2} (C^n_{bt}(f^b_t - bias^{obs}_f) + g^n)dt^2 \tag{1}$$

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Standard Inertial Prediction

$p_t = p^n_{t1}$
$v_t = v^n_{t1}$
$\phi_t = \phi^n_{t1}$
for $t1 < t < t2$ **do**
  $\Delta t = t_{t+1} - t_t$
  $f^n_t = C^n_{bt}\left(f^b_t - bias_f\right)$
  $v_{t+1} = v_t + f^n_t \Delta t + g^n \Delta t$
  $p_{t+1} = p_t + v_t \Delta t$
  $\phi_{t+1} = \phi_t + E^n_{bt}\left(\omega^b_t - bias_\omega\right)\Delta t$
end for

$$\text{prediction} = \begin{bmatrix} p_t \\ v_t \\ \phi_t \end{bmatrix}$$

Algorithm 2 Standard Inertial Prediction Covariance Calculation

$P_t = P_{t1}$
for $t1 < t < t2$ **do**
  $\Delta t = t_{t+1} - t_t$
  $\alpha = \frac{dC^n_{bt}\left(f^b_t - bias_f\right)}{d\phi_t}$
  $\beta = \frac{dE^n_{bt}\left(\omega^b_t - bias_\omega\right)}{d\phi_t}$

$$F_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3\Delta t & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \alpha\Delta t & -C^n_{bt}\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + \beta\Delta t & \mathbf{0}_3 & -E^n_{bt}\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$$

$$G_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ C^n_{bt}\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & E^n_{bt}\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

  $P_{t+1} = FP_tF' + GQG'$
end for
$P_{t2} = P_t$

$$v^n_{t2} = v^n_{t1} + \int_{t1}^{t2} (C^n_{bt}(f^b_t - bias^{obs}_f) + g^n)dt \tag{2}$$

$$\phi^n_{t2} = \phi^n_{t1} + \int_{t1}^{t2} E^n_{bt}(\omega^b_t - bias^{obs}_\omega)dt. \tag{3}$$

As IMUs provide discrete time samples of the accelerations and rotation rates experienced, these equations are used in their discrete form. Algorithm 1 shows an example implementation if used for state prediction in an extended Kalman filter (EKF).

For SLAM applications, both the integrated inertial navigation solution and the estimate of the uncertainty of that solution are required. The uncertainty is needed so that when observations from other sensors are combined with the inertial observations, they can be weighted to give the optimal estimate of the vehicle states.

Algorithm 2 shows how the uncertainty of the inertial navigation solution, i.e., P_{t2}, can be calculated given the initial vehicle state uncertainty P_{t1}, and the IMU sensor observation noise covariance matrix Q.

TABLE I
DERIVATIVES OF THE INERTIAL PREDICTION EQUATIONS WITH RESPECT TO ESTIMATED STATES

	p^n_t	v^n_t	ϕ^n_t	$bias^{obs}_f$	$bias^{obs}_\omega$	g^n
p^n_{t+1}	I	$I\Delta t$	$\frac{1}{2}\frac{dC^n_{bt}}{d\phi^n_t}\left(f^b_t - bias^{obs}_f\right)\Delta t^2$	$-\frac{1}{2}C^n_{bt}\Delta t^2$	0	$\frac{1}{2}I\Delta t^2$
v^n_{t+1}	0	I	$\frac{dC^n_{bt}}{d\phi^n_t}\left(f^b_t - bias^{obs}_f\right)\Delta t$	$-C^n_{bt}\Delta t$	0	$I\Delta t$
ϕ^n_{t+1}	0	0	$\mathrm{I} + \frac{dE^n_{bt}}{d\phi^n_t}\left(\omega^b_t - bias^{obs}_\omega\right)\Delta t$	0	$-E^n_{bt}\Delta t$	0

C. Inertial Simultaneous Localization and Mapping Initialization

The navigation implementations that are discussed are computed in a globally referenced frame. The inertial body frame and visual observations are transformed into this frame before being fused, and therefore, the initial condition requirement still exists. This is shown in Algorithm 1, where the requirement for p^n_{t1}, v^n_{t1}, and ϕ^n_{t1} is provided at the beginning.

Table I shows the derivatives of the standard inertial integration equations [see (1)–(3)] with respect to the vehicle states. It can be seen from this table that the majority of the nonlinearity comes from the vehicle attitude terms (with some nonlinearity with respect to the IMU bias terms as well). This is because of the nonlinear sine and cosine terms in the rotation and rotation rate matrices.

The position and velocity states have a perfectly linear interaction and, therefore, do not pose a problem for a linear estimator.

The major problem is with unknown initial attitude due to the nonlinear effects it has on the estimate. This causes errors when a linear estimator, such as an EKF, is used, and slow convergence or even instability when a linearized batch solver is used.

In [3], the initial orientation and attitude of the unmanned aerial vehicle was obtained by keeping it stationary on the runway, while GPS observations were made of its position, and internal tilt sensors were used to get the roll and pitch. The use of such an initialization routine is common, but for the situations that are considered in this paper, it is inconvenient to the user.

In [13], the aircraft would fly with GPS and inertial observations being made over a period of time until sufficient information was available for the navigation solution to converge. This large initial angle uncertainty technique, when applied with an EKF or other nonrelinearizing implementation, suffers from accumulated linearization errors from uncertainties in the vehicle states. These are normally handled by adding additional stabilizing noise to the filter by inflating the observation or process model uncertainties. This procedure is often unstable for high-dynamic applications, like the one considered in this paper, where the initial attitude has a large range of possible values, not just close to flat and level for aircraft applications.

As a result, a batch initialization routine is probably the best suited to the target application as it does not require the user to perform a specialized initialization routine, and it provides a more numerically stable and accurate initialization.

III. INERTIAL PREINTEGRATION THEORY

Regardless of the initialization procedure chosen, all the IMU observations are traditionally transformed into the navigation

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

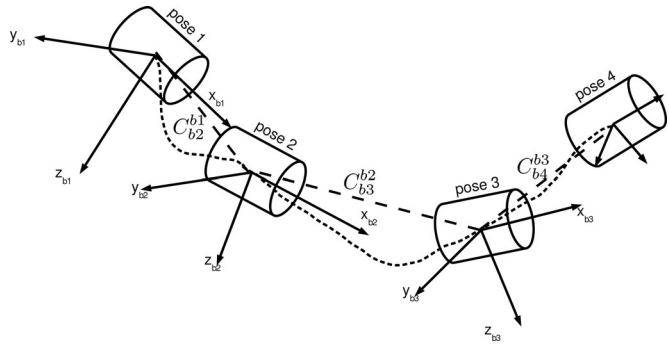

Fig. 2. Inertial integration in the body frame of the last pose of interest.

frame before integration. IMUs are sampled at very high rates when compared with other sensors that are used in navigation, on the order of hundreds of samples per second, even in low-cost units.

This requires updates to be performed at high rates for a marginalizing filter, such as the EKF, or a large number of pose states being required for delayed state methods. Another problem this causes, specifically for batch initialized inertial methods, is that it requires a large number of inertial observations that are to be stored and processed in the batch filter once the initial conditions become observable.

If these observations could be integrated first without knowing the initial conditions of the vehicle, then a number of inertial observations could be treated as a single observation in the filter, reducing the problems that are listed previously.

One possible way to do this is to integrate the inertial observations between required poses in the body frame of the previous pose, as presented in [16]. An illustration of this concept is shown in Fig. 2, where the frame that is used for integration of the inertial observations moves along with the vehicle from pose to pose.

Since in many navigation applications, poses are only required at the rate of the next fastest sensor other than the IMU, for example, the frame rate of the camera, and this sensor usually takes samples at a much lower rate, many inertial observations can be integrated between poses this way.

If the inertial integration equations (1)–(3) are rewritten to perform the integration in the body frame of the last pose, the following equations are obtained:

$$p_{t2}^n = p_{t1}^n + (t2-t1)v_{t1}^n + \iint_{t1}^{t2} g^n dt^2 + C_{bt1}^n \iint_{t1}^{t2} \left(C_{bt}^{bt1}\left(f_t^b - bias_f^{obs}\right)\right) dt^2 \quad (4)$$

$$v_{t2}^n = v_{t1}^n + \int_{t1}^{t2} g^n dt + C_{bt1}^n \int_{t1}^{t2} \left(C_{bt}^{bt1}\left(f_t^b - bias_f^{obs}\right)\right) dt \quad (5)$$

$$\phi_{t2}^n = \phi_{t1}^n + E_{bt1}^n \int_{t1}^{t2} E_{bt}^{bt1}\left(\omega_t^b - bias_\omega^{obs}\right) dt. \quad (6)$$

The initial conditions for the rotation matrix C_{bt}^{bt1} at the start of the integration period time $t1$ is C_{bt1}^{bt1}, which is the Identity matrix.

These equations still provide the vehicle pose estimates in the globally referenced navigation frame, but the integration of the inertial observations between poses is performed in the body frame of the last pose and then transformed into the navigation frame after integration instead of before.

One thing to note from (4)–(6) is that the integrations are performed in the vehicle body frame; the vehicle states with respect to this frame can be perfectly known. As a result, the inertial observations can actually be integrated with no initial condition requirements, and even before the states themselves are estimated.

If the integrals of the inertial observations from (4)–(6) are extracted, the following equations are obtained:

$$\Delta p_{t2}^{+t1} = \iint_{t1}^{t2} C_{bt}^{bt1}\left(f_t^b - bias_f^{obs}\right) dt^2 \quad (7)$$

$$\Delta v_{t2}^{t1} = \int_{t1}^{t2} C_{bt}^{bt1}\left(f_t^b - bias_f^{obs}\right) dt \quad (8)$$

$$\Delta \phi_{t2}^{t1} = \int_{t1}^{t2} E_{bt}^{bt1}\left(\omega_t^b - bias_\omega^{obs}\right) dt. \quad (9)$$

These terms that can be preintegrated without initial conditions represent the change in position, velocity, and attitude of the vehicle from pose 1 to pose 2 in the (moving) body frame of pose 1.

These preintegrated sets of observations can then be used as a single delta state observation in place of all the IMU observations that occur between these two poses. Therefore these integrated terms will be referred to as preintegrated inertial delta observations.

Once calculated these delta components can then be substituted back into (4)–(6) as in (10)–(12), shown below. In these equations the integration of the gravity term has also been simplified, which can be done as the gravity vector integrand contains no time-dependent terms (the $\frac{1}{2}$ factor in (10) is a byproduct of the double integration process)

$$p_{t2}^n = p_{t1}^n + (t2-t1)v_{t1}^n + \frac{1}{2}(t2-t1)^2 g^n + C_{bt1}^n \Delta p_{t2}^{+t1} \quad (10)$$

$$v_{t2}^n = v_{t1}^n + (t2-t1)g^n + C_{bt1}^n \Delta v_{t2}^{t1} \quad (11)$$

$$\phi_{t2}^n = EulerFromDCM\left(C_{bt1}^n \Delta C_{bt2}^{bt1}\right). \quad (12)$$

The delta attitude component ΔC_{bt2}^{bt1} is multiplied by the previous attitude rotation matrix and then converted back into the Euler representation that is used for state estimation. This is done to avoid the small-angle approximation that is used by the Euler rotation rate matrix as this may no longer be valid for the longer integration integrals used. Furthermore, the delta attitude component refers to an actual change in attitude over

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

the integration interval and not a rotation rate, as is the case with the raw gyro observations.

A. Δp^+ Component

The delta attitude ΔC_{bt2}^{bt1} and delta velocity Δv_{t2}^{t1} components represent a change in the estimated states directly; however, an observation of the change in position cannot be obtained from the inertial observations alone. As the accelerometer observation only provides acceleration measurements, the initial velocity of the vehicle at the start of the integration interval is required.

As the integrations are performed without any knowledge of the initial vehicle states, an integration reference frame located at, and moving with, the previous body pose was used. Since the reference frame is moving at the same instantaneous velocity as the vehicle was at the previous pose, an initial velocity with respect to this frame of zero can be used in the preintegrated position delta (7), making integration without initial conditions possible.

This, however, means that when the preintegrated inertial delta observations are transformed back into the navigation frame to predict the current vehicle pose, not only is a translation and rotation required, but the difference in reference frame velocities needs to be accounted for as well.

Given the velocity estimate at the start of the integration period, a simple constant velocity estimate of the new position can be calculated. This new position estimate then forms the reference frame transformation.

This form of constant velocity model is common in SLAM when no other information is available; however the final position of the platform will be dependent on the acceleration profile over the integration period as well.

To take this acceleration profile into account, the Δp^+ component is used. It can be seen as a corrective term that is added to the constant velocity prediction of the new vehicle position to account for acceleration. The "$+$" superscript is used to distinguish it from a true delta position term and to indicate that it is an additive correction.

In (10), the constant velocity prediction can be seen in the $(t2-t1)v_{t1}^n$ component with the Δp^+ correction in the $C_{bt1}^n \Delta p_{t2}^{+t1}$ component.

B. Bias Correction

From the delta component in (7)–(9), it can be seen that the initial velocity and attitude estimates are no longer required to perform inertial observation integration. However the corrective terms for the IMU sensor biases are still present.

If an estimate of the biases is present at the time of integration, then they can be used in these calculations. However, if they are not available or if they are inaccurate and are refined at a later time, it would be useful to be able to correct for these biases without having to reperform the integrations.

This is possible if the derivatives of the preintegrated inertial delta observation components that are shown in (7)–(9) with respect to the IMU biases are calculated. This will be shown in Section IV-D.

Algorithm 3 Inertial Delta Observation Creation

$\Delta p_t^+ = 0$
$\Delta v_t = 0$
$\Delta \phi_t = 0$
for $t1 < t < t2$ **do**
  $\Delta t = t_{t+1} - t_t$
  $f_t^{bt1} = C_{bt}^{bt1}\left(f_t^b - bias_f^{obs}\right)$
  $\Delta v_{t+1} = \Delta v_t + f_t^{bt1}\Delta t$
  $\Delta p_{t+1}^+ = \Delta p_t^+ + \Delta v_t \Delta t$
  $\Delta \phi_{t+1} = \Delta \phi_t + E_{bt}^{bt1}\left(\omega_t^b - bias_\omega^{obs}\right)\Delta t$
end for
$$\text{observation} = \begin{bmatrix} \Delta p_t^+ \\ \Delta v_t \\ \Delta \phi_t \end{bmatrix}$$

IV. Implementation of Preintegrated Inertial Delta Component Calculations

The inertial preintegration technique that is described in Section III generates a three-component pseudo-observation from a number of raw IMU observations. These pseudo-observations are referred to as preintegrated inertial delta observations.

A. Inertial Delta Observation Creation

The process of generating preintegrated inertial delta observations from (7)–(9) is shown in Algorithm 3. This can be compared with the standard technique to process IMU observations for SLAM that is shown in Algorithm 1.

Note how the standard technique requires the initial position, velocity, and attitude estimates for the platform, whereas the proposed technique sets all these values to zero for purposes of the integration. Because of this difference, the proposed technique integrates the inertial observations in the frame of the body at the start of the integration interval instead of in the navigation frame.

In addition, in the standard technique, the gravity vector needs to be accounted for when integrating the inertial observations in the velocity equation. For the proposed technique, gravity does not need to be considered until after the observations are already integrated when the navigation frame state estimates are required.

B. Covariance and Jacobian Calculation

The covariance calculations for preintegrated inertial delta observations are shown in Algorithm 4, which can be compared with the standard technique in Algorithm 2.

Similar to the case for the observation creation, the proposed technique sets the initial uncertainty to zero and performs calculations in the frame of the body at the start of the integration period, while the standard technique required an initial state uncertainty estimate and performs calculations in the navigation frame.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

Algorithm 4 Inertial Delta Jacobian and Covariance Creation

$J_t = \mathbf{I}_{15}$
$R_t = \mathbf{0}_{15}$
for $t1 < t < t2$ **do**
$\Delta t = t_{t+1} - t_t$
$\alpha = \frac{dC_{bt}^{bt1}\left(f_t^b - bias_f^{obs}\right)}{d\phi_t}$
$\beta = \frac{dE_{bt}^{bt1}\left(\omega_t^b - bias_\omega^{obs}\right)}{d\phi_t}$

$$F_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3\Delta t & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \alpha\Delta t & -C_{bt}^{bt1}\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + \beta\Delta t & \mathbf{0}_3 & -E_{bt}^{bt1}\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$$

$$G_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ C_{bt}^{bt1}\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & E_{bt}^{bt1}\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

$J_{t+1} = F_t J_t$
$R_{t+1} = F_t R_t F_t' + G_t Q G_t'$
end for
$J_{t1}^{t2} = J_t$
$R_{t1}^{t2} = R_t$

One major difference with the preintegrated inertial delta observations technique is the requirement for an additional term, i.e., the Jacobian of the observation uncertainty, J_{t1}^{t2}. This Jacobian is required if a correction to the bias estimate is to be made after the inertial observations are already preintegrated, as will be described in Section IV-D.

C. Inertial Delta Observation Use

Once the preintegrated inertial delta components have been created, they can be incorporated into a navigation solution just like any other sensor observation. If they are to be used in the filter prediction stage, such as in standard EKF inertial SLAM [3], then the prediction equations (10)–(12) can be used.

If, however, an information space filter method is used where they are treated as observations, then the state prediction equations need to be rearranged to provide a prediction of the preintegrated inertial delta observation given the estimated states. Equations (10)–(12) can be rearranged to give the preintegrated inertial delta observations for a given set of vehicle positions and velocities. Given the current estimates of recent vehicle positions, velocities, gravity, and sensor biases, an expectation of the preintegrated inertial delta components with respect to these state estimates can be taken, giving

$$\begin{aligned} E\left(\Delta p_t^+ | \hat{x}_t\right) = \widehat{C}_n^{bt1}\left(\widehat{p}_{t2} - \widehat{p}_{t1} - \widehat{v}_{t1}\Delta t - \frac{1}{2}\hat{g}\Delta t^2\right) \\ + \frac{d\Delta p_t^+}{dbias_f}\left(\widehat{bias}_f - bias_f^{obs}\right) \\ + \frac{d\Delta p_t^+}{dbias_\omega}\left(\widehat{bias}_\omega - bias_\omega^{obs}\right) \end{aligned} \tag{13}$$

$$\begin{aligned} E\left(\Delta v_t | \hat{x}_t\right) = \widehat{C}_n^{bt1}\left(\widehat{v}_{t2} - \widehat{v}_{t1} - \hat{g}\Delta t\right) \\ + \frac{d\Delta v_t}{dbias_f}\left(\widehat{bias}_f - bias_f^{obs}\right) \\ + \frac{d\Delta v_t}{dbias_\omega}\left(\widehat{bias}_\omega - bias_\omega^{obs}\right) \end{aligned} \tag{14}$$

$$\begin{aligned} E\left(\Delta\phi_t | \hat{x}_t\right) = EulerFromRotationMatrix(\widehat{C}_n^{bt1}\widehat{C}_{bt2}^n) \\ + \frac{d\Delta\phi_t}{dbias_\omega}\left(\widehat{bias}_\omega - bias_\omega^{obs}\right) \end{aligned} \tag{15}$$

where $E(.)$ is the expected value operator, and $\hat{x}_t$ is the current estimate of the mean of the vehicle states.

D. Postintegration Bias Correction

The $\frac{d\Delta.}{dbias}$ terms in (13)–(15) are the derivatives of the inertial delta observation components with respect to the IMU biases. They are used to correct for any changes in the estimated IMU biases, since the preintegrated inertial delta observations were formed.

The $\widehat{bias}$ terms are the current estimates of the IMU biases, and the $bias^{obs}$ terms are the bias values that are used to create the preintegrated inertial delta observations.

The derivatives of the inertial delta observation components with respect to the IMU bias terms are contained within the preintegrated inertial delta observations Jacobian matrix J that was calculated in Algorithm 4. The components of this matrix can be seen in

$$J = \begin{bmatrix} \frac{d\Delta p_{t2}^+}{dp_{t1}^{t1}} & \frac{d\Delta p_{t2}^+}{dv_{t1}^{t1}} & \frac{d\Delta p_{t2}^+}{d\phi_{t1}^{t1}} & \frac{d\Delta p_{t2}^+}{dbias_f} & \frac{d\Delta p_{t2}^+}{dbias_\omega} \\ \mathbf{0}_3 & \frac{d\Delta v_{t2}}{dv_{t1}^{t1}} & \frac{d\Delta v_{t2}}{d\phi_{t1}^{t1}} & \frac{d\Delta v_{t2}}{dbias_f} & \frac{d\Delta v_{t2}}{dbias_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \frac{d\Delta\phi_{t2}}{d\phi_{t1}^{t1}} & \mathbf{0}_3 & \frac{d\Delta\phi_{t2}}{dbias_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{dbias_f}{dbias_f} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{dbias_\omega}{dbias_\omega} \end{bmatrix} \tag{16}$$

V. Body Referenced Navigation Frame

In Sections III and IV, a method to integrate a number of inertial observations in the body frame of the vehicle was developed. This method allows the integration to be performed without the need for the initial conditions of the vehicle pose to be known.

However, these initial conditions are still required after integration when the created delta observations are fused into a navigation solution, as shown in Section IV-C. This was because of the use of a globally referenced navigation frame.

Globally referenced navigation frames are traditionally used in inertial navigation applications as the desired navigation solution and the aiding observations, such as when GPS is used.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

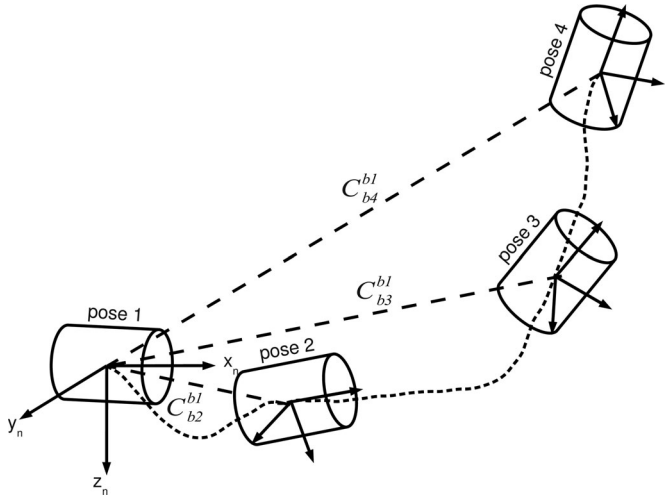

Fig. 3. Inertial navigation in a first pose referenced navigation frame.

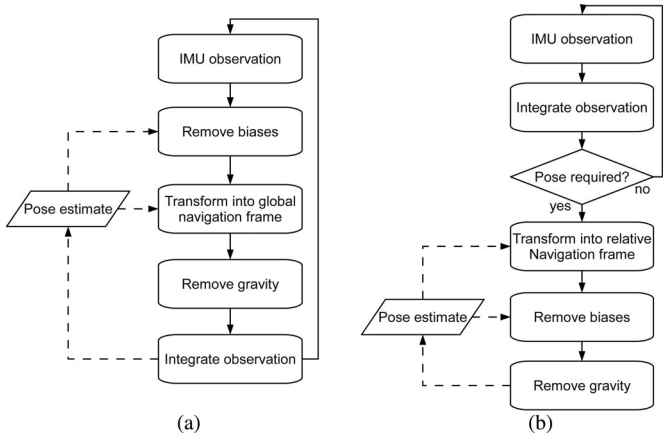

Fig. 4. Flowcharts comparing the steps in standard inertial integration with those of the reparameterized form. (a) Standard inertial integration process, as done in [3]. (b) Reparameterized inertial integration to improve linearity.

This requires the position and orientation of the vehicle in this globally referenced frame to be known.

For the applications that are considered in this paper, operating in and around buildings, all observations are made in a body referenced frame. In addition, for most SLAM applications, a navigation frame relative to the starting position is sufficient, or even necessary if no absolute vehicle pose information is available.

The main problem with the uncertainty about the initial vehicle pose in the selected navigation frame is the nonlinear effect of vehicle orientation through the rotation matrices. However, if instead a stationary navigation frame that is fixed to the body frame of the first position of the vehicle is used (i.e., a frame that is not moving at the instantaneous velocity of the vehicle at that point as was the case in Section III), then the initial attitude of the vehicle can be known with complete certainty. This removes the nonlinear effects of the vehicle orientation from the required initial conditions.

Fig. 3 shows an illustration of inertial navigation in a navigation frame fixed to the first vehicle pose. This can be compared with Fig. 1 for the globally referenced case.

The advantage of using a body referenced navigation frame is that the initial attitude, which is nonlinear, is now known. However, as a consequence of using an arbitrarily oriented navigation frame, the gravity vector in this navigation frame is now unknown.

Since the gravity vector in the body frame is initially unknown, even though the initial attitude is fixed, there are still the same number of unknown variables with body frame parameterization. The advantage of this method is in the fact that the gravity vector estimate is linear with respect to the estimated states, whereas the attitude is not.

VI. Overview of the Proposed Inertial Integration Technique

The body referenced inertial navigation frame and preintegration technique allows the ordering of the steps involved in using inertial observations in a navigation or SLAM implementation to be changed. These changes provide advantages in terms of linearity and initialization requirements for the system.

Fig. 4 shows two flowcharts comparing the standard method for using inertial observations for aided navigation as used in [3] with the one proposed in this paper. Notice how the steps involved are the same; only the ordering and the stages at which estimated states are required has changed.

A. Initial Condition Recovery

Because of the reparameterization of the navigation frame and the preintegration of the inertial observations, all the required initial conditions of the vehicle now become linearly dependent on the estimated states. Therefore, if a batch initialization is to be used, it can be performed with a simple linear estimator instead of using nonlinear methods for the standard technique. This has advantages for both speed and stability of the initialization.

However, since all the initial conditions are linearly dependent, they can actually be directly estimated in a linear filter, such as when implemented in an EKF. This has great advantages for visual-inertial-aided navigation implementations, since a special initialization stage is not required. This is similar to the large-angle uncertainty initialization methods for inertial navigation that is mentioned in Section II, except now, the nonlinearity has been removed.

Regardless of which initialization method is chosen, it is useful to know how many observations are required before the initial conditions can be known. Since the navigation frame is fixed to the position and orientation of the vehicle at the first pose, only the initial velocity and the gravity vector remain as unknown initial conditions.

B. Initial Velocity Observability

In order to investigate how the initial velocity of the vehicle becomes observable, the preintegrated inertial delta observation state prediction equations can be inspected.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

By rearranging (10), it is possible to show that the initial velocity of the platform is observable, given two consecutive relative position estimates, a gravity vector estimate, and the preintegrated inertial delta observations between them. This is shown as follows:

$$v_{t1}^n = \frac{p_{t1}^n - p_{t2}^n + C_{bt1}^n \Delta p_{t2}^{+t1} + \frac{1}{2}(t2-t1)^2 g^n}{(t2-t1)}. \quad (17)$$

Therefore, given the gravity vector in the navigation frame and a relative position estimate between two poses from another aiding sensor, such as a stereo camera, the initial velocity of the platform can be obtained in a linear way without the need for a prior estimate.

C. Gravity Vector Observability

One criterion for the initial velocity observability that is shown in Section VI-B is that an estimate of the gravity vector in the navigation frame has to be available. For the inertial preintegration process to be truly free from prior initial condition requirements, the gravity vector should be recoverable in a linear manner as well.

Taking (10) over two consecutive inertial delta observations and the velocity equation from the first delta observation from (11) results in

$$\begin{aligned} p_{t2}^n &= p_{t1}^n + (t2-t1)v_{t1}^n + \frac{1}{2}(t2-t1)^2 g^n + C_{bt1}^n \Delta p_{t2}^{+t1} \\ p_{t3}^n &= p_{t2}^n + (t3-t2)v_{t2}^n + \frac{1}{2}(t3-t2)^2 g^n + C_{bt2}^n \Delta p_{t3}^{+t2} \\ v_{t2}^n &= v_{t1}^n + (t2-t1)g^n + C_{bt1}^n \Delta v_{t2}^{t1}. \end{aligned} \quad (18)$$

Substituting the velocity equation into the second position equation gives

$$\begin{aligned} p_{t3}^n &= p_{t2}^n + (t3-t2)\left(v_{t1}^n + (t2-t1)g^n + C_{bt1}^n \Delta v_{t2}^{t1}\right) \\ &\quad + \frac{1}{2}(t3-t2)^2 g^n + C_{bt2}^n \Delta p_{t3}^{+t2} \\ &= p_{t2}^n + (t3-t2)\left(v_{t1}^n + C_{bt1}^n \Delta v_{t2}^{t1}\right) + C_{bt2}^n \Delta p_{t3}^{+t2} \\ &\quad + (t3-t2)\left(\frac{1}{2}(t3-t2) + (t2-t1)\right) g^n. \end{aligned} \quad (19)$$

Then, substituting (17) into (19) gives an equation in terms of the relative position estimates, inertial delta observation components, and the gravity vector alone. This can be rearranged into an expression for the gravity vector in (20), shown at the bottom of this page.

Therefore, with inertial preintegration and a body referenced navigation frame, it is possible to linearly obtain estimates for both the initial velocity of the platform and the gravity vector after relative position estimates between just three poses. As this whole initial condition recovery process is linear, no prior estimates of the initial conditions are required to use as starting points for the estimation.

A beneficial side effect of the gravity vector observability is that with this estimate, the absolute roll and pitch of the vehicle in the inertial frame can be extracted as well.

This derivation also shows that the gravity vector does not become observable until images from at least three poses are available. Three poses are needed in order to be able to distinguish between gravity and acceleration. If IMU observations are taken from just one pose, the acceleration that the vehicle is undergoing is not observable; therefore, gravity cannot be estimated without making assumptions about this acceleration. After two poses are observed, the initial velocity can be obtained but it takes a third pose observation before the average acceleration of the vehicle can be observed from the images, and this way, the acceleration component can be separated out from the gravity component of the accelerometer readings.

A simple filter could be used at this point to estimate the gravity vector after the third image and initialize the IMU roll and pitch for use in a standard inertial SLAM implementation. It will be shown in Section VIII-D that even though there is a gravity estimate available at this point, it is not accurate, and therefore, there are still linearization problems until more observations are obtained and estimate convergence is achieved.

VII. Algorithm Representation and Implementation

A graphical representation of the SLAM problem in information space based on the graphical SLAM [6], [7] formulation was chosen for the filter implementation to test this system. As this filter keeps past poses in the filter instead of marginalizing them out as in normal EKF-based SLAM, it falls in the class of delayed state filters.

Graphical SLAM has several advantages over traditional EKF-based SLAM filters. These include the ability to relinearize past observations and to reassess data association, as well as being able to add and remove individual states and observations at any time.

The ability to reassess data association is very useful for this application as the visual feature matching often includes a number of outliers that cannot be identified until after they are already included in the solution. The ability to relinearize the nonlinear visual observation functions also has a significant effect on the quality of the estimates that are obtained.

A. Edge Energy Outlier Detection

Visual features are extracted using a Harris corner detector and then matched using a pyramidal implementation of Lucas–Kanade (LK) optical flow. This method was chosen as it is fast and performs well in the environment used. It also keeps the visual feature data association process independent from the

$$g^n = \frac{p_{t2}^n - p_{t3}^n + (t3-t2)\left(\left(\frac{p_{t1}^n - p_{t2}^n + C_{bt1}^n \Delta p_{t2}^{+t1}}{(t2-t1)}\right) + C_{bt1}^n \Delta v_{t2}^{t1}\right) + C_{bt2}^n \Delta p_{t3}^{+t2}}{(t3-t2)\left(\frac{1}{2}(t3-t2) + \frac{3}{2}(t2-t1)\right)}. \quad (20)$$

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

state estimation so that errors in the estimated states do not affect the observations made.

Harris corners was chosen since it is fast and gives well-localized and distinct features. A feature descriptor is not needed for the optical flow algorithm to work, and features are not needed to be reacquired for the navigation method used; therefore, a robust descriptor such as scale-invariant feature transform (SIFT) [14] or Speeded Up Robust Feature (SURF) [1] is not required. Even though these methods locate features using extrema in the second derivative of image intensity similar to how Harris corners work with the exception that for SIFT and SURF, the features must be a local extrema in scale space as well. Larger scale features are not of use for this navigation application as they have a larger location uncertainty in the image plane and, therefore, do not contribute as much information to the camera position estimate as the smaller scale features do.

The feature selection threshold for the feature detector is automatically adjusted at each image to extract between 20 and 50 features per image. This level was chosen as platform pose estimation does not improve significantly with additional visual features after the first 20 features have already be added. Additional features after this point only slows down the estimation process without adding much in terms of accuracy. Not all of the extracted features are successfully matched to adjacent frames; however, the estimation process is robust enough to handle these situations.

The fundamental matrix [10] between consecutive images is calculated within a RANdom SAmple Consensus (RANSAC) [5] routine to detect and reject outliers in the LK optical flow feature matched. This method works well to identify the majority of the outliers, but it is not 100% effective as any match along an epipolar line will be retained, even if these matches are not consistent over multiple frames.

After observations have been added to the filter, the data association process can be tested at any time by analyzing the graph edge energy [6] of the visual observation in question. The observation edge energy can be thought of as being similar to a normalized innovation conditioned on the current state estimates. If it is too large ($>2\sigma$), for a given observation, this indicates that the observation is not consistent with the current estimates of the state means. Therefore, there is a high probability of the data association for this observation being incorrect.

This second test for visual data association is particularly useful during periods where there are not a sufficient number of tracked features between frames for the fundamental matrix that is to be calculated.

B. Sliding Window Forced Independence Smoothing

When implementing a visual-inertial navigation solution, it is desirable to retain past vehicle poses and feature locations in the filter, as is done in graphical SLAM. However, since loop closure is not performed in this implementation, new observations made have little effect on the estimate of past vehicle poses and the features observed from them. As retaining these states in the filter consumes both memory and processing time, it would be desirable to be able to remove them from the filter when they no longer have much influence on the current pose estimate.

This can, traditionally, be done either by conditioning on the old states or by marginalizing them out. If conditioning is performed, the past poses and feature locations have their estimated means fixed, and thereby, the current state estimates would be conditioned on those that have been fixed. Alternatively, with marginalizing, the linearizations in the past observations would be fixed, and then, old states would be marginalized out. The problem with both these methods is that they can lead to overconfident estimates of the current states, and if a filter error occurred in the past, such as incorrect data association or poor linearization of observations, then these errors will propagate throughout the whole trajectory.

A modification to the graphical SLAM implementation that has been made for the navigation filter used in this application is the complete removal of past observation information and state estimates from the current solution after a fixed time. When an estimated vehicle pose is more than a set amount of time old, it is removed from the filter, as well as all the observations made from this pose. If a landmark has had all its observations removed from this action, the landmark is removed from the filter as well. As a result, observations and estimates made far in the past have no influence on the current vehicle pose or feature location estimates at all.

We call this procedure "sliding window forced independence" (SWFI) as the window of poses currently estimated in the filter slides through time as new poses are added and the removal of past observations and states forces the current estimate to be independent of these previous observations and estimates. This differs from conditioning or marginalizing out the past poses, as with SWFI, the past observations and estimates have no influence at all on the current estimate. This is only possible because of the preintegrated inertial delta observations not requiring any prior initial condition information since otherwise, the initial condition estimates would have to be derived from the previous filter estimates.

The main advantage of using SWFI is that it isolates errors in the filter to the region about where they occur. This method also inherently incorporates the ability of the filter to recover automatically from any errors as soon as the cause of the error has left the filtering window. As an additional bonus, this method bounds the computational load to a constant time as only the observations from a fixed number of poses are being processed at any one time.

If the whole trajectory, not just the poses in the sliding window, need to be represented in a common reference frame, this can be achieved by setting the initial position and orientation of the oldest pose in the window to be equal to the position and orientation of the pose that was just removed. As the initial velocity and gravity vector are reestimated at each stage from only the observations that are made within the sliding window at all times, this redefining of the navigation frame maintains the independence of the current estimate from the previous observations. This is possible because the initial orientation and position of the reference frame used with preintegrated inertial delta observations has no effect on the estimation procedure,

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

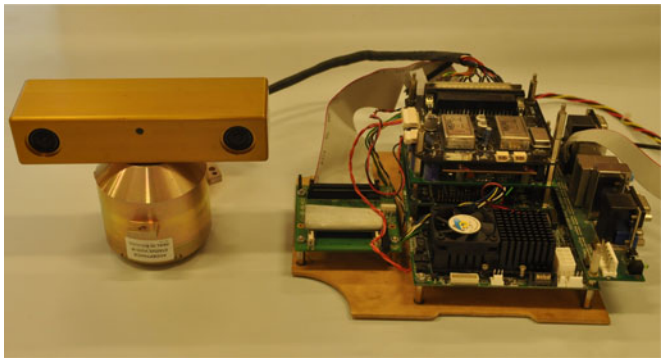

Fig. 5. Sensor suite used to obtain the walking datasets. The Point Grey Research Bumblebee2 stereo camera can be seen on the top of the unit with the Honeywell HG1900 IMU mounted just below it.

TABLE II
HG1900 IMU MEASURED SPECIFICATIONS

Sampling rate	600 Hz
Accelerometer noise	$0.0775\ \text{ms}^{-2}(1\sigma)$
Gyroscope noise	$0.001\ \text{rads}^{-1}(1\sigma)$
Accelerometer bias stability	$\pm 0.003\ \text{ms}^{-2}$
Gyroscope bias stability	$\pm 6.0e^{-5}\ \text{rads}^{-1}$

and the inertial integration is still performed in the body fixed frame.

For the experiments that are presented in this paper, a sliding window size of 30 image poses was used.

VIII. EXPERIMENTAL RESULTS

In order to test the performance of the preintegrated inertial observation method presented in this paper in real-world situations, datasets were collected on a human-mounted sensor suite walking in and around buildings. This setup was used to obtain the type of data that would be expected from a first responder or highly dynamic autonomous system.

A. Sensors

The sensor suite contains a Honeywell HG1900 IMU, which provides inertial observations at 600 Hz and a Point Grey Research Bumblebee2 stereo camera unit that has been fitted with 2.1-mm wide angle lenses. The wide angle lenses were used to make it possible to obtain more useful observations in confined areas, such as hallways and staircases, where there are usually few distinct visual features. Wide angle lenses also provide increased parallax for landmarks observations when moving forward. Stereo images were recorded at a frame rate of 6.25 Hz.

Fig. 5 shows a photograph of the sensor configuration that is used with the stereo camera mounted on top and the IMU just below it. Tables II and III show the specifications for the IMU and cameras, respectively. This unit was hand held and carried around the buildings for datasets that are collected.

The stereo camera pair was calibrated for lens distortion and optical alignment using the Camera Calibration Toolbox for MATLAB [2]. Alignment between the cameras and IMU, as well as the IMU biases, was estimated online within the filter implemented.

TABLE III
POINT GREY RESEARCH BUMBLEBEE2 WITH 2.1-MM LENSES SPECIFICATIONS

Sampling rate	6.25 Hz
Focal length	2.1 mm
Resolution	640×480 pixels
Field of view	$97^o \times 80^o$
Angular resolution	0.2^o (at centre)
Stereo baseline	12 cm

B. Navigation Filter Implementation

The states that are estimated in the navigation filter are shown in (21). For each vehicle pose, the position, velocity, and orientation are estimated, except for the first pose for which only velocity is estimated as the navigation frame is fixed to its position and orientation. The IMU to camera offset and rotation as well as the gravity vector and IMU biases and landmark locations for currently tracked features are also estimated

$$[p_n, v_n, \phi_n, \ldots, p_2, v_2, \phi_2, v_1 \; g, \phi_{IMU}^{camera}, p_{IMU}^{camera}, bias_f, bias_\omega, \; l_1, \ldots l_m]'. \quad (21)$$

As the sliding window progresses and new poses are added to the state vector, the oldest pose is removed and the map is now conditioned on the position and orientation estimate of the next oldest pose. Landmark states are also removed from the state vector when they are no longer observed by any current poses.

C. System Validation

In order to test that the inertial preintegration and the system as a whole was performing as expected, a simple validation experiment was conducted.

In a normal office environment, two locations were marked on the ground 9 m apart. The sensor suite was then carried back and forth between these two points, first going forward then going backward. Whenever one of these points was reached, the unit was placed on the ground over the point so that the true location and velocity of the unit at this time were known and compared with the navigation solution estimate.

Fig. 6 shows a sample image from the datasets that are used in this paper. The area shown in this image is where the system validation test was conducted.

The resultant estimated trajectory of the unit for this test can be seen in Fig. 7. The estimated pose states at the marked reference locations are shown in Table IV. The first pose is exactly zero for the position and attitude estimates as this pose is used to define the navigation frame.

The two points are 9 m apart, and it can be seen from pose 80 and 290 that the estimated location is within a few centimeters of this value. The solution does appear to drift slightly with the location of the sensor platform at the final pose (pose 430) being estimated as 16 cm from the original starting position. This is expected since no loop closure is performed, and lost visual features are not reacquired.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

Fig. 6. Sample image taken from one of the datasets used in this paper. This image shows the kind of indoor environment in which experiments were conducted.

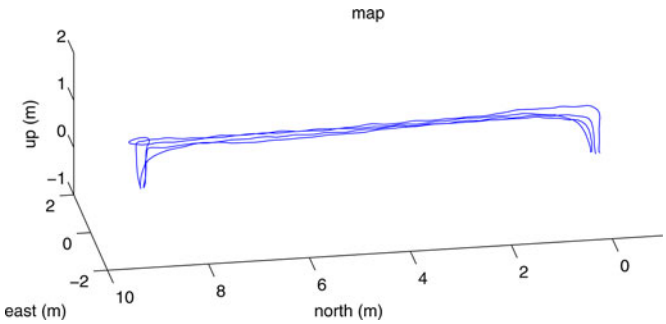

Fig. 7. Estimated trajectory of the sensor platform when moving back and forth between two marked locations on the ground that are 9 m apart. The north, east, and up directions are in the body frame of the first pose and are not globally referenced.

TABLE IV
ESTIMATED POSITION, VELOCITY, AND ATTITUDE OF THE SENSOR PLATFORM DURING THE STATIONARY PERIODS AT THE MARKED LOCATIONS IN THE DATASET SHOWN IN FIG. 7

Pose number	1	80	186	290	430
North position (m)	0	9.0653	0.0698	8.9882	-0.1074
East position (m)	0	-0.3422	0.0578	-0.2286	0.1142
Down position (m)	0	-0.0090	-0.0239	0.0284	0.0293
Roll (^{o})	0	-0.1432	0.0802	-0.0115	0.0401
Pitch (^{o})	0	0.0000	-0.6646	-0.7792	0.0516
Yaw (^{o})	0	0.3209	-179.7655	179.6796	0.6761
North velocity (ms^{-1})	0.0046	-0.0090	0.0001	-0.0002	-0.0009
East velocity (ms^{-1})	0.0047	0.0094	-0.0080	0.0040	-0.0015
Down velocity (ms^{-1})	0.0033	-0.0040	0.0086	-0.0081	-0.0080

The estimated velocities of this trajectory are very accurate with each of the components of the velocity at each stationary location being estimated at less than 1 cm·s^{-1}. Each of the attitude components are also estimated to less than 1° of the true value at each stationary position. This is within the accuracy of placement of the unit as is expected.

Table V shows the estimated gravity vector at the times corresponding to the stationary poses in Table IV. As these poses are more than 30 images apart and the SWFI window size is 30 poses, these estimates are completely independent; however, they are all in the frame of the first pose; therefore, they can be compared.

TABLE V
ESTIMATED GRAVITY VECTOR IN THE FIRST POSE FRAME CORRESPONDING TO THE PERIODS IN TABLE IV

Pose number	1	80	186	290	430
North component (ms^{-2})	0.0181	0.0241	0.0194	0.0153	0.0356
East component (ms^{-2})	-0.0308	-0.0333	-0.0374	-0.0374	-0.0443
Down component (ms^{-2})	9.7959	9.7970	9.8016	9.7987	9.7981
Magnitude (ms^{-2})	9.7860	9.7970	9.8016	9.7988	9.7983

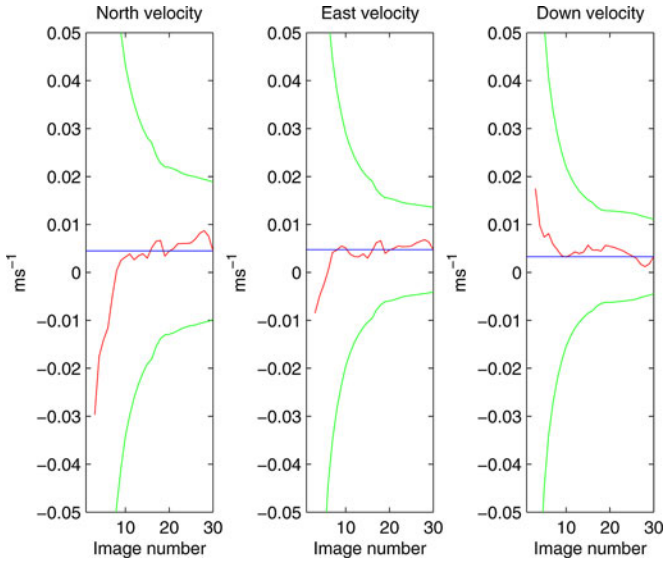

Fig. 8. Estimate of the components of the initial velocity over time as observations are added to the solution with the 2σ uncertainty bounds. The blue line represents the final value. Only results for the first 30 poses are shown as this is the size of the sliding window used.

The estimated gravity magnitudes are within 0.06% of each other with a maximum angular deviation of 0.12°. The average estimated gravity vector magnitude of 9.796 m·s^{-2} is comparable with the true gravity magnitude value for Sydney of 9.797 m·s^{-2}.

The true gravity vector magnitude is not supplied to the filter, and no gravity magnitude constraint is applied; this value has been derived purely from the observations made.

D. Initial Condition Estimation

In Section VI-A, the observability of the initial conditions for the inertial navigation solution, those being the initial instantaneous velocity and gravity vector in the navigation frame, was discussed. It was shown that the initial conditions become linear observable after three sets of visual observations are made.

Figs. 8 and 9 show how the estimates of these initial conditions and their predicted uncertainty evolves over time for the test dataset that is presented in Section VIII-C for the initial velocity and gravity vector estimate respectively.

The first meaningful estimates of these values appear after observations from the third set of images are incorporated into the filter. This is as expected from the analysis that is performed in Section VI-A, which is why the estimates that are shown in Figs. 8 and 9 start at image number 3.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

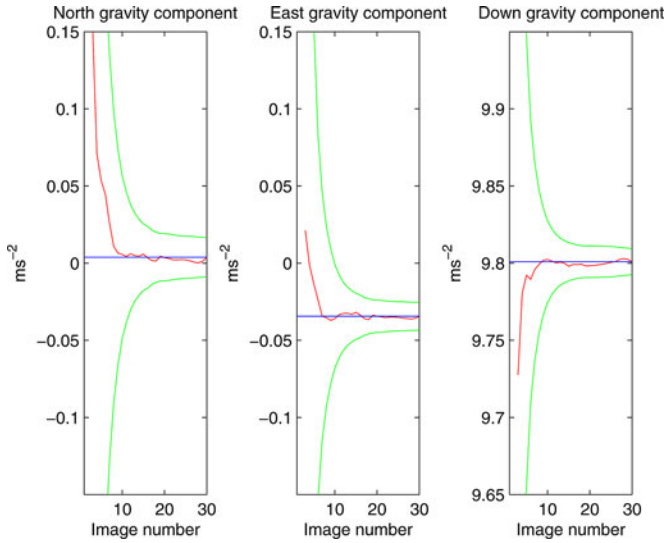

Fig. 9. Estimate of the components of the gravity vector over time as observations are added to the solution with the 2σ uncertainty bounds. The blue line represents the final value. Only results for the first 30 poses are shown as this is the size of the sliding window used.

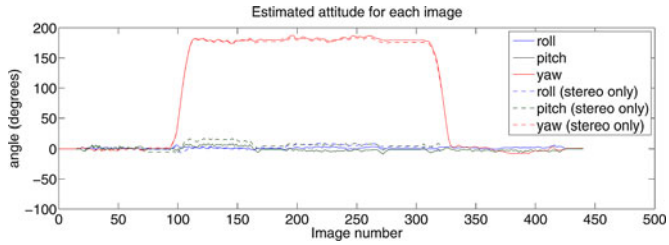

Fig. 10. Estimated components of the platform attitude for the dataset shown in Fig. 7 for the stereo-inertial and the stereo-only case for comparisons. The stereo-only estimate lost track of the platform position at image 321 when the angular velocity of the platform was high and there was little overlap between images.

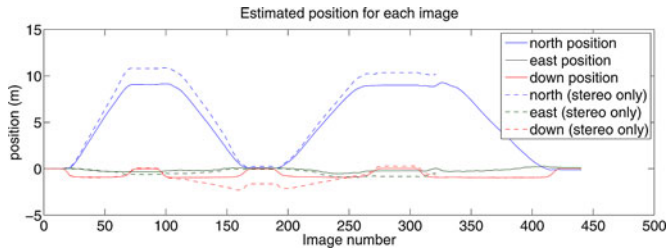

Fig. 11. Estimated components of the platform position for the dataset shown in Fig. 7 for the stereo-inertial and the stereo-only case for comparison. The stereo-only estimate lost track of the platform position at image 321 when the angular velocity of the platform was high and when there was little overlap between images.

Notice how the initial uncertainty for these estimates is large, but it converges quickly over time. In addition, the final uncertainty for the velocity estimate components are on the order of ± 1 cm$\cdot$s^{-1}, which is in agreement with the estimated velocities at the stationary positions shown in Table IV.

The final uncertainty in the north (forward) components of the initial conditions is also larger than for the east (right) and down components. This is as expected because of the forward looking cameras used which provide less accurate position estimates in the forward direction.

As an SWFI window size of 30 poses is used, only the estimates for the first 30 poses of the dataset are shown. From this point onward, old observations are removed from the filter as new images are added; therefore, further convergence of the initial condition estimates is not expected to occur.

E. Comparison With Stereo-Only Estimation

In order to assess the benefit of adding inertial observations to the stereo observations, the validation dataset that is shown in Fig. 7 was processed using only the stereo camera observations for comparison with the stereo-inertial result. This comparison can be seen in Figs. 10 and 11.

Fig. 10 shows the estimated attitude of the platform during this dataset for both the stereo-inertial and stereo-only cases. It can be seen from this figure that the stereo-only case closely follows the stereo inertial estimate of the attitude in yaw, but some difference can be seen in the pitch and roll estimates that do not vary as much.

One problem with the stereo-only result is that at image number 321, it loses track of the location of the platform (this can be seen more clearly in Fig. 10). During this time, the platform is rotating at a high rate, approximately 90° per second, which results in little overlap between images. This low amount of overlap means that the feature tracker is unable to match features between image number 321 and 322 to constrain the pose of the platform from vision alone. The stereo-inertial case does not have a problem here as the inertial observations are able to constrain the drift until visual features are acquired again.

In Fig. 11, a further problem with the stereo-only case is evident. The second position that the platform stops at is 9 m from the starting position as described in Section VIII-C; however, the stereo-only estimate of this pose is between 10.4 and 10.8 m from the starting point.

Fig. 6 gives a clue as to why the distance estimate is off by so much. It can be seen from this image that the area in which the dataset was taken is a fairly open part of a large room. As the stereo baseline of the cameras used is only 12 cm, the range estimates that are obtained from the cameras are only accurate up to about 4 or 5 m. As most of the features in this room are at the outer limit of accuracy for these cameras, the range to the visual features and, therefore, the scale of the camera motions can not be estimated accurately. The inertial observations help in this case by measuring the acceleration of the platform and implicitly fixing the scale as a result.

F. Real-World Datasets

After the system had been tested, two longer datasets of walking around an office building, as well as through and between two adjacent buildings, were taken. Fig. 12 shows an external view of these two buildings, and the area that is used has been highlighted. These datasets simulate the kind of motion and observations that would be obtained from such a localization system mounted on a first responder.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

Fig. 12. Buildings that the real-world datasets were collected in and around. Both levels of the Rose St. building, as well as the third floor of the Link building, were used and have been highlighted. The bridge between the two buildings, as well as the staircase on the far right of the Link building, and the parking lot were also traversed.

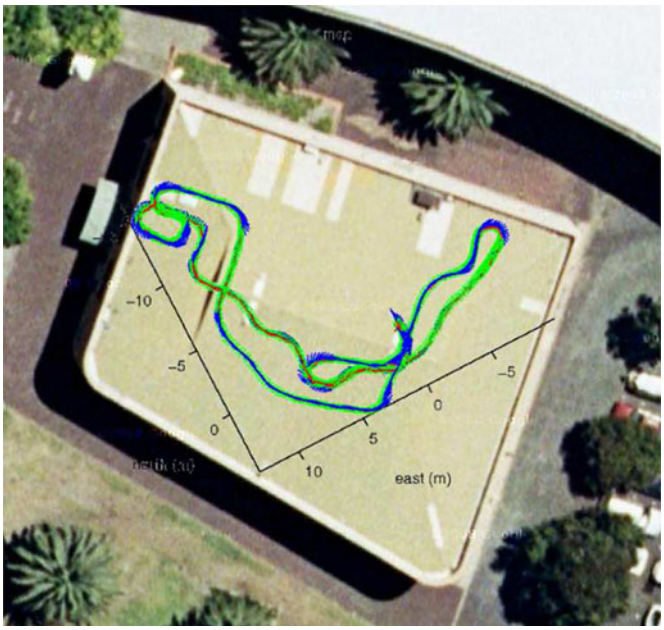

Fig. 13. Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building overlaid onto a satellite image of the area. The roll and pitch of the map was aligned using the estimated gravity vector, and then, the yaw was rotated by hand to align with the photo.

The first dataset walking around the two stories of the Rose St. building at The University of Sydney can be seen in Fig. 13 from above, overlaid on a satellite photo of the area as a side view of the estimated trajectory in Fig. 14. This dataset contains observations from 713 image pairs that are taken over 114 s and covers a distance of approximately 120 ms.

Similar results for the second dataset starting at the same location but then walking up stairs and across a bridge to the adjacent Link building then outside and back to the starting location are shown in Figs. 15 and 16. This dataset contains observations from 1130 image pairs taken over 180 s and covers a distance of approximately 180 m.

In both these datasets, estimation was started while moving at an unknown velocity with an unknown orientation to demonstrate how the system can start in any configuration and is not

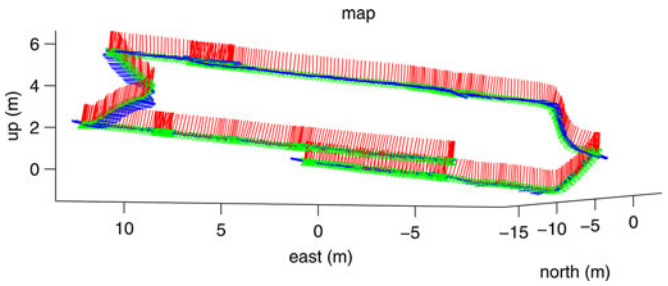

Fig. 14. Side view of the estimated trajectory of the first dataset showing the two levels of the Rose St. building. The red cross at (0,0) is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The back set of stairs can be clearly seen on the right side of the image as well as the misalignment of the two portions of the trajectory on the bottom level.

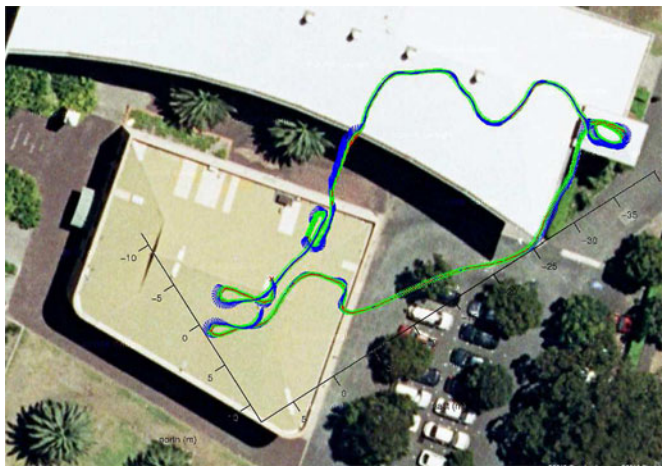

Fig. 15. Top view of the estimated trajectory of the second dataset between the Rose St. building and the Link building overlaid onto a satellite image of the area. The trajectory does not line up exactly partly due to the viewpoint of the image not being taken from directly above and the drift in the solution toward the end of the trajectory. The bridge between the two buildings is not present in this photo.

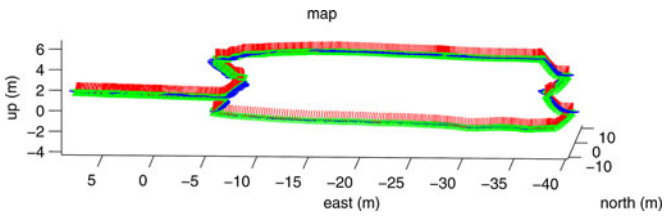

Fig. 16. Side view of the estimated trajectory of the second dataset showing the two levels of the Rose St. building and the staircase down the back of the Link building. The red cross at (0,0) is the starting point. The red lines show the up direction at each image pose, and the blue lines are the forward direction at these poses. The Link building stair case can be clearly seen on the right side of the figure.

required to be stationary. This is important for the application considered as it is desirable for no specialized initialization routine or conditions that are to be required.

The final pose in the datasets is at approximately the same location as the initial pose in both cases; however, the orientation was different as the unit was traveling in the opposite direction. A comparison of the estimates of the first and last pose from the

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
FIRST AND LAST POSE ESTIMATES FOR THE TRAJECTORY IN THE FIRST ROSE ST. BUILDING DATASET

Pose number	1	713
North position (m)	0	-0.02909
East position (m)	0	0.26414
Down position (m)	0	0.78309
Roll (^{o})	0	5.57339
Pitch (^{o})	0	5.80399
Yaw (^{o})	0	-156.105
North velocity (ms^{-1})	0.70932	-0.87437
East velocity (ms^{-1})	0.14298	-0.19654
Down velocity (ms^{-1})	-0.08262	-0.05031

TABLE VII
FIRST AND LAST POSE ESTIMATES FOR THE TRAJECTORY IN THE SECOND DATASET THROUGH THE ROSE ST. BUILDING AND THE LINK BUILDING

Pose number	1	1130
North position (m)	0	0.99440
East position (m)	0	2.27144
Down position (m)	0	0.11145
Roll (^{o})	0	1.99135
Pitch (^{o})	0	20.5475
Yaw (^{o})	0	-172.399
North velocity (ms^{-1})	0.76594	-1.07630
East velocity (ms^{-1})	0.15865	-0.11597
Down velocity (ms^{-1})	0.15966	-0.07324

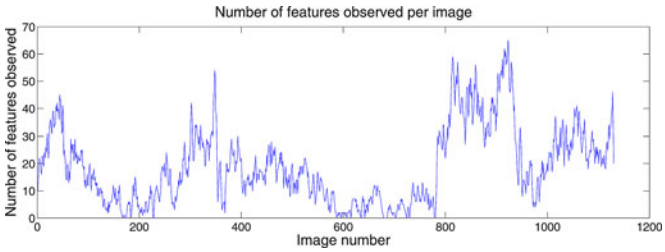

Fig. 17. Number of visual observations made per image for the second dataset between the Rose St. and Link buildings. Notice the periods of few observations around image number 200 and between image numbers 600 to 800 when the unit was inside the building stair cases.

first dataset within the Rose St. building can be seen in Table VI, with a similar comparison for the second dataset between the Rose St. and the Link buildings shown in Table VII.

The final error in estimated position is approximately 0.82 m or 0.68% of distance traveled for the first dataset and 2.5 m or 1.39% of distance traveled for the second dataset.

The larger error in the final position for the second dataset can be explained by looking at Fig. 17. This figure shows the number of visual observations that are obtained from each image in the dataset. It can be seen that this number drops at around image number 200, as well as between image numbers 600 to 800.

The period between images 600 and 800 corresponds to the time that the sensor unit was inside the staircase of the link building, which can be seen as the part of the trajectory in the far right of Fig. 15, as well as the vertical part in the far right of Fig. 16.

During this long period of few visual observations, the filter has to rely more on the observations from the IMU to constrain the estimate of the navigation solution, which drifts over time and results in the larger final estimated position error for this dataset.

G. Scale Estimation With Monocular Observations

The results presented so far used observations from a stereo camera as this gives better conditioned results as well as observing the true scale of the environment. If only a single camera is used for SLAM [4], then the true scale of the environment is not observable because of the bearing-only nature of projective cameras.

However, it was shown in [15] that with the addition of accelerometer observations and sufficient linear acceleration, the true scale of the environment can be recovered.

Using observations from the left camera only and a similar technique to [15] where a weak range prior was applied between the first two poses until the solution converged (in this case, after ten image pairs were observed), a monocular-inertial navigation solution was estimated for the first dataset through the Rose St. building (similar results were obtained for the second dataset).

As the weak range prior was removed after the tenth image was added to the filter, the final scale is completely a product of the IMU and left camera observations alone. A comparison of the estimated trajectories for the stereo-inertial case already presented and the monocular-inertial case where the scale of the environment has been estimated from the accelerometer observations can be seen in Fig. 18.

Fig. 18 shows that the general shape of the trajectory is similar for both cases; however, the estimate of the final position is not as accurate. In the monocular-inertial case, the final position error is approximately 2.9 m compared with 0.82 m for the stereo-inertial case. One area of the trajectory that may be the cause of the majority of the error is the straight portion from approximately $(-12, 6)$ to $(-7, 8)$, which looks elongated in the monocular-inertial case.

Fig. 19 shows a comparison of the estimated distance between poses for the monocular and the stereo case. The portion of the trajectory that is mentioned previously occurs around the 340 image mark. It can be seen from Fig. 19 that the estimated distance between the poses is much larger for the monocular case in this area, which confirms the elongation of this portion of the trajectory.

Fig. 20 shows the ratio of these two distance estimates, and the peak around image 340 confirms this. In this part of the building, there were few close features as it was in the middle of a seminar area which would have lead to less accurate relative position estimates from the visual observations. The resultant increase in the error of the estimated translation is the result of this larger position uncertainty.

It can be seen from Fig. 20 that the mean ratio of the estimated scales for the monocular and the stereo map is around 1, which is as expected for a well-estimated monocular scale. The spikes in the scale ratio around image 570 occurs when the sensor unit is inside the back staircase, and few visual observations are present to constrain the position estimate.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

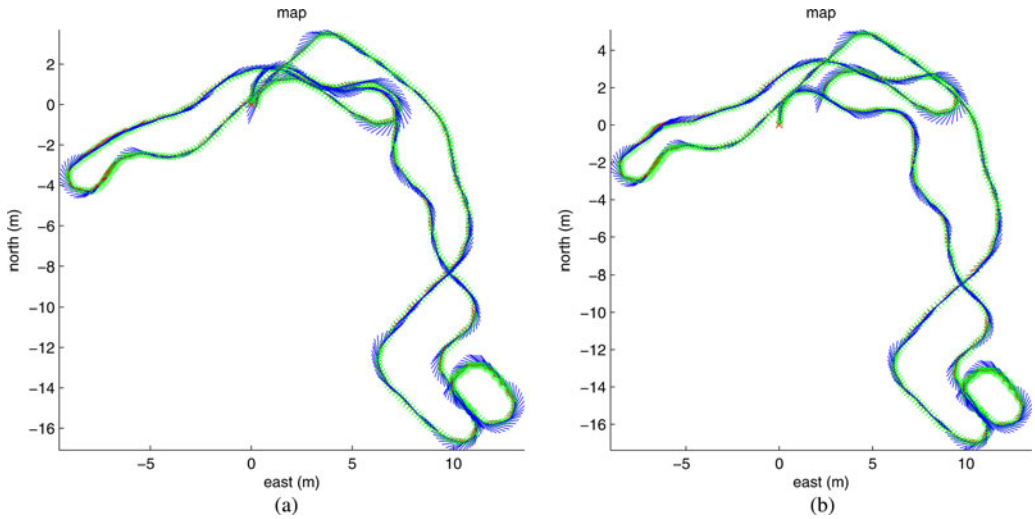

Fig. 18. Comparison of the navigation solution for the stereo-inertial case with the monocular-inertial case, where the map scale is estimated from the accelerometer observations. (a) Top view of estimated trajectory for the stereo-inertial navigation solution. (b) Top view of estimated trajectory for the monocular-inertial navigation solution.

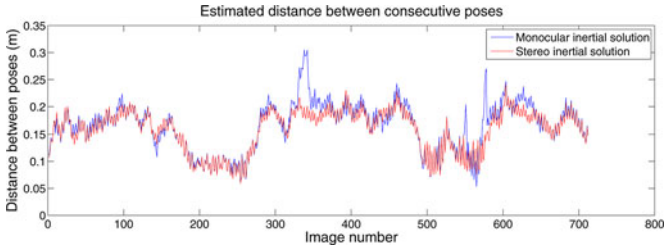

Fig. 19. Comparison between the estimated distance between poses for the stereo-inertial and monocular-inertial case.

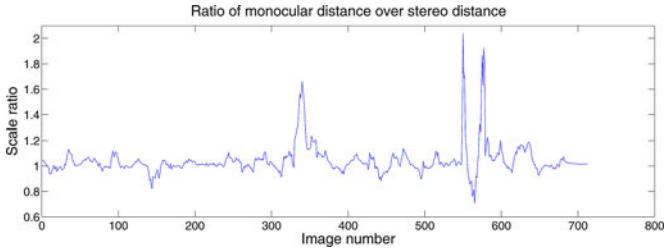

Fig. 20. Ratio of the estimated distance between poses for the stereo-inertial and monocular-inertial case.

IX. Conclusion and Future Work

A. Conclusion

In this paper, a vision-inertial navigation system for use in and around buildings in an urban environment has been presented. The inertial integration equations were reparameterized into a body referenced frame, instead of the usual globally referenced frame used for standard inertial navigation.

This transformation of the reference frame for inertial navigation removed the uncertainty about the initial orientation of the platform with respect to the navigation frame at the expense of creating uncertainty about the gravity vector in the navigation frame. This transfer of uncertainty allows for the initial conditions of the inertial integration to be recovered quickly in a linear way, removing the need for a special initialization procedure.

It was also shown to be possible to integrate a number of inertial observations in the body frame of the IMU between required poses without any knowledge of the state of the system. This allows a large number of high rate IMU observations to be combined into a single observation, making it faster and easier to deal with in a SLAM or navigation filter.

This system was implemented using a graphical SLAM filter, where all information from past poses and observations were removed from the filter after a set amount of time using SWFI. The advantages of this method, while providing a slightly suboptimal solution, include isolating errors in the solution to an area local to where they occurred and stopping the continual accumulation of errors throughout the solution. The computational complexity of the solution is also bounded to constant time.

Results from a dataset of the implemented system moving between known locations were shown to validate the performance of the system. The observability and convergence of the estimated initial conditions with no prior information was demonstrated and confirmed from experimental data. The estimation of the gravity magnitude within 0.06%, as well as the velocity of the system under $1\,\mathrm{cm}\cdot\mathrm{s}^{-1}$ error in the indoor environment was also demonstrated.

Two longer datasets demonstrating the performance of the system in the real-world target application were also shown. For these examples, position uncertainties on the order of 1% of distance traveled was obtained even with prolonged periods of few visual observations.

Finally, the addition of an IMU to a monocular SLAM implementation in order to make the true scale of the environment observable as shown in [15] was also demonstrated. One of the longer indoor datasets taken over two stories of an office building was processed using observations from the IMU and only a single camera and compared with the stereo camera result.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.

The results showed that even though the stereo result was more accurate, the monocular result was very similar and the true environment scale was being accurately estimated.

B. Future work

Future work includes a more rigorous study of the possible failure modes of this system. Automatic detection of these failure modes would also be desirable as they could inform the user that navigation accuracy may be compromised.

Detection of loop closure on revisiting a previously explored area would also be a benefit for this system. This could be used to adjust for long-term drift in the navigation solution as well as to combine information from multiple users operating in the same area.

REFERENCES

[1] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[2] J. Y. Bouguet. Camera calibration toolbox for MATLAB. (Jul. 9, 2010). [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc

[3] M. Bryson and S. Sukkarieh, "Building a robust implementation of bearing-only inertial SLAM for a UAV," *J. Field Robot.*, vol. 24, no. 1–2, pp. 113–143, 2007.

[4] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," presented at the Int. Conf. Comput. Vis., Nice, France, Oct. 2003.

[5] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[6] J. Folkesson and H. I. Christensen, "Robust SLAM," in *Proc. IFAC Symp. Intell. Auton. Veh.*, 2004.

[7] J. Folkesson and H. I. Christensen, "Graphical SLAM for outdoor applications," *J. Field Robot.*, vol. 24, pp. 51–70, 2007.

[8] J. Folkesson and H. I. Christensen, "SIFT based graphical SLAM on a Packbot," in *Proc. Int Conf. Field Service Robots*, 2007, pp. 317–328.

[9] P. Gemeiner, A. Davison, and M. Vincze, "Improving localization robustness in monocular SLAM using a high-speed camera," presented at the Robot.: Sci. Syst. IV Conf., Zurich, Switzerland, Jun. 2008.

[10] R. I. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *ECCV '92: Proceedings of the Second European Conference on Computer Vision*. London, U.K.: Springer-Verlag, 1992, pp. 579–587.

[11] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proc. IEEE/RSJ Int. Conf. Proc. Intell. Robots Syst.*, Sep. 2008, pp. 3946–3952.

[12] J. H. Kim and S. Sukkarieh, "Airborne simultaneous localization and map building," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 406–411.

[13] X. Kong, "Inertial navigation system algorithms for low cost IMU," Ph.D. dissertation, Univ. Sydney, Sydney, Australia, 2000.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[15] T. Lupton and S. Sukkarieh, "Removing scale biases and ambiguity from 6 DoF monocular SLAM using inertial," in *Proc. Int. Conf. Robot. Autom.*, 2008, pp. 3698–3703.

[16] T. Lupton and S. Sukkarieh, "Efficient integration of inertial observations into visual SLAM without initialization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1547–1552.

[17] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "A constant time efficient stereo SLAM system," in *Proc. Brit. Mach. Vis. Conf.*, 2009.

[18] L. M. Paz, P. Pinies, J. D. Tardos, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 946–957, Oct. 2008.

[19] B. B. Ready and C. N. Taylor, "Inertially aided visual odometry for miniature air vehicles in GPS-denied environments," *J. Intell. Robot. Syst.*, vol. 55, no. 2–3, pp. 203–221, 2009.

[20] B. M. Scherzinger, "Inertial navigator error models for large heading uncertainty," in *Proc. IEEE Position Location Navigat. Symp.*, Apr. 1996, pp. 477–484.

[21] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. Washington, DC: Amer. Inst. Aeronautics Astronautics, 2004.

Todd Lupton received the B.E. degree in mechatronics and the Ph.D. degree in robotics from The University of Sydney, Sydney, N.S.W., Australia, in 2005 and 2010, respectively.

He is currently with Silverbrook Research, Balmain, N.S.W.

Salah Sukkarieh received the B.E. degree in mechatronics and the Ph.D. degree in robotics from The University of Sydney, Sydney, N.S.W., Australia, in 1997 and 2000, respectively.

He is currently an Associate Professor with The University of Sydney, where he is also the Research Director with the Australian Centre for Field Robotics.

Authorized licensed use limited to: Beijing Xiaomi Technology Co.Ltd. Downloaded on April 24,2022 at 13:07:46 UTC from IEEE Xplore. Restrictions apply.