THE JOURNAL OF NAVIGATION, Page 1 of 19. © The Royal Institute of Navigation 2014
doi:10.1017/S0373463314000848

Fusing Stereo Camera and Low-Cost Inertial Measurement Unit for Autonomous Navigation in a Tightly-Coupled Approach

Zhiwen Xian, Xiaoping Hu and Junxiang Lian

(*Department of Automatic Control, College of Mechatronics and Automation, National University of Defense Technology*)
(E-mail: xphu2012.nudt@gmail.com)

Exact motion estimation is a major task in autonomous navigation. The integration of Inertial Navigation Systems (INS) and the Global Positioning System (GPS) can provide accurate location estimation, but cannot be used in a GPS denied environment. In this paper, we present a tight approach to integrate a stereo camera and low-cost inertial sensor. This approach takes advantage of the inertial sensor's fast response and visual sensor's slow drift. In contrast to previous approaches, features both near and far from the camera are simultaneously taken into consideration in the visual-inertial approach. The near features are parameterised in three dimensional (3D) Cartesian points which provide range and heading information, whereas the far features are initialised in Inverse Depth (ID) points which provide bearing information. In addition, the inertial sensor biases and a stationary alignment are taken into account. The algorithm employs an Iterative Extended Kalman Filter (IEKF) to estimate the motion of the system, the biases of the inertial sensors and the tracked features over time. An outdoor experiment is presented to validate the proposed algorithm and its accuracy.

KEYWORDS

1. Autonomous. 2. Multi-sensor navigation. 3. Robust estimation. 4. Imaging

Submitted: 25 May 2014. Accepted: 13 November 2014.

1. INTRODUCTION. Autonomous navigation for mobile vehicles is a popular current topic (Yun et al., 2013; Chowdhary et al., 2013). It is vital to know a vehicle's position, velocity and attitude in many applications. An Inertial Measurement Unit (IMU) consisting of three orthogonal accelerometers and gyroscopes is able to track a carrier's motion with high frequency, and has been successfully used for vehicle navigation. However, the accuracy of the IMU deteriorates with time due to the accumulation of inertial sensor biases and noises (Titterton and Weston, 2004; Noureldin et al., 2011). GPS can provide position and velocity with limited error. The accumulated error of an inertial sensor will receive periodic correction by integrating IMU and GPS. Unfortunately, in environments where GPS signals are unobtainable

(e.g., indoor, forest, underwater, on Mars, etc.) the GPS-aided IMU system is not suitable. Furthermore, high precision GPS receivers are always expensive and bulky, which are not suitable for certain applications.

An alternative approach to restrain IMU error is the use of visual sensors such as stereo cameras. Some important advantages of the Visual-Inertial System (VIS) are listed below:

- The VIS can be lightweight, low-cost and is smaller than a high accuracy IMU/GPS integrated system. With rapid recent development, IMUs and cameras have become smaller and cheaper and are more common (such as in cars and mobile phones).
- Since both IMUs and cameras do not need the transmission or reception of any radio signals, they are completely passive sensors and are thus able to be used in a GPS-denied environment and be part of an autonomous navigation system.
- The VIS is able to produce a more robust and accurate motion estimation than either a camera or IMU acting alone. The main advantage of the IMU is that it is able to accurately track the motion of a rapidly changing vehicle over a short time, but it is subject to low frequency drift. In contrast, visual motion estimation is more accurate when the camera is moving slowly. These complementary properties work together to give a better motion estimation.

Recently, the fusion of vision and inertial sensors for navigation has received considerable attention in the research community. Corke et al. (2007) presented a tutorial introduction of inertial and visual sensing from a biological and an engineering perspective. Several algorithms of relative pose (translation and rotation) calibration for hybrid inertial/visual systems can be found in Lang and Pinz (2005), Lobo and Dias (2007) and Mirzaei and Roumeliotis (2008). Much research (Armesto et al., 2007; Veth and Raquet, 2007; Kelly and Sukhatme, 2011; Chowdhary et al., 2013) has been conducted into fusion algorithms of inertial and visual sensors for navigation. However, these studies do not pay much attention to the different type of features in an unknown environment.

In an unknown environment, the features may be near or far from the camera. The near feature provides both distance and orientation while the far one mainly offers orientation information. In this paper, we present our work on combining a stereo camera and a low-cost inertial sensor for navigation in an Iterative Extended Kalman Filter (IEKF) estimator. The features are divided into two categories, one is far or viewed only in one camera, and the other can be seen in both cameras and has a high parallax. To represent the two kinds of features, the Inverse Depth point (introduced in Civera et al. (2008)), and 3D Cartesian point is employed.

In order to make full use of the VIS's potential, the following factors have also been considered: the varying biases of the IMU; the initial attitude of the IMU with respect to gravity and feature detection, initialisation, tracking and management. A real data experiment has been carried out to evaluate the performance of the proposed algorithm. The result demonstrates that the algorithm presented in this paper is able to navigate in an unknown environment autonomously, and has better accuracy than visual-only and inertial-only approaches.

The remainder of the paper is organised as follows: related work is examined in Section 2. In Section 3, we introduce our system and briefly discuss the preliminaries

of the paper. In Section 4, we describe the VIS model, and then develop our IEKF-based estimator in Section 5. An outdoor experiment and result are given in Section 6. Finally, we draw the main conclusions of our work in Section 7.

2. RELATED WORK. A considerable number of studies have been done on both visual and inertial navigation. Visual odometry is particularly relevant to our work, which has been focused on the use of either monocular or stereo vision to estimate the egomotion of an agent from the environment. For monocular vision, the detected feature has scale unobservability. In order to overcome this problem, Davison et al. (2007) and Feng et al. (2012) used the fixed depth constraint. A delay feature initialisation scheme were presented in Davison (2003) and Kim and Sukkarieh (2003). The stereo camera, however, is able to provide scale through the baseline between cameras. Davison (Davison, 1998; Davison and Murray, 2002) demonstrated an active stereo visual Simultaneous Localisation and Mapping (SLAM) system based on EKF, but they did not consider the distant features, so that it can be only used in the near scene, such as indoors (Se et al., 2002). Distant features have bearing information and it is unwise not to use them (Paz et al., 2008). However, vision-only techniques depend on the available features, so it is difficult to recover the real track when all tracked features are lost. In order to overcome the limitations of vision-only techniques an approach that integrates a stereo camera and an IMU is developed.

Considerable work has been reported recently about a hybrid stereo camera and inertial system. In order to estimate an Unmanned Aerial Vehicle's (UAV's) position and velocity, Carrillo et al. (2012) and Kelly and Saripalli (2008) used a Kalman Filter to fuse stereo visual odometry and inertial measurements. As a loosely coupled approach, they did not use the inertial sensors to predict the tracked features. Our approach incorporates stereo images and inertial measurements in a tight model. Veth and Raquet (2007) developed an image-aided inertial navigation algorithm, which is implemented by using a multi-dimensional stochastic feature tracker. The algorithm is specifically evaluated for operation using low-cost, CMOS imagers and Micro-Electro-Mechanical Systems (MEMS) inertial sensors. The principal drawback of this algorithm is that the number of landmarks actively tracked has to be constant, so the excess tracked features are wasted. Mourikis and Roumeliotis (2007) presented an EKF-based algorithm for vision-aided inertial navigation. The author introduced a special measurement model that does not require the 3D feature position in the state vector. However, this approach needs to store all the tracked features, and requires more storage capacity, which makes the algorithm possibly not available in certain applications. While all of these algorithms estimate motion and structure, they do not consider the different type of features, (i.e. near and far features), which is the primary focus of our work. Furthermore, we account for the IMU biases and the alignment of the IMU with respect to local gravity.

3. PRELIMINARIES. In this section, we introduce our system used for autonomous navigation. Then three reference frames and some notation that are used throughout the remainder of the paper are presented. Finally, two kinds of feature point parameterisations are introduced.

3.1. System Overview. Our VIS consists of a stereo camera and a MEMS IMU (MIMU) sensor, as shown in Figure 1. The visual sensor, a PointGrey

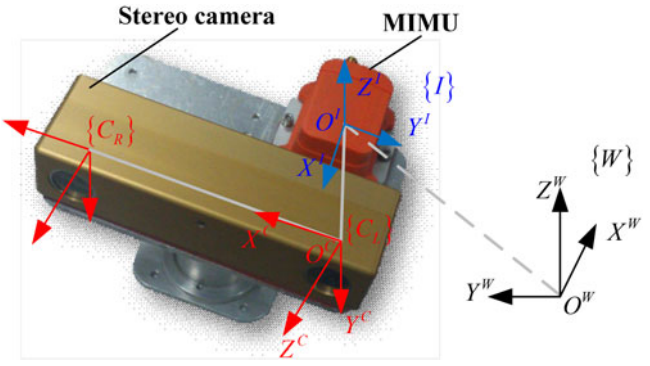


Figure 1. The VIS which consists of the stereo camera and the MIMU and the relationship between the world $\{W\}$, camera $\{C_L\}$, $\{C_R\}$, and IMU $\{I\}$ reference frames. The stereo camera and the MIMU are rigidly attached.

Bumblebee2 stereo camera, is able to provide a corrected stereo image, so we need not to consider the lens distortions and camera misalignments. The inertial sensor is a MEMS-based Mti-G unit, manufactured by Xsense Technologies, and it can provide three-axis angular rate, linear acceleration and earth magnetic field data. To synchronize the camera and the MIMU, a discrete bus is utilised. Due to the MIMU and camera being attached rigidly, the relative pose of the camera and the IMU, which were carefully calibrated before using the sensors, can be seen as constant.

3.2. Reference Frames and Notation. When working with a sensor unit containing a stereo camera and MIMU, we consider three reference frames:

1) The world frame $\{W\}$: The pose of the VIS is estimated with respect to this frame that is fixed to the earth. The features of the scene are modelled in this frame. It can be aligned in any way; however, in this paper it is vertically aligned, and with the x, y, z-axes aligned with the north, west and vertical axes respectively.
2) The camera frame $\{C\}$: This frame is attached to the moving stereo camera, with its origin at the optical centre of the camera, and with the z-axis pointing along the optical axis. There are two camera frames, as shown in Figure 1, where they are aligned with each other with a known translation (in our case, the direction cosine matrix between frame C_L and frame C_R is an identity matrix, and the length of the baseline is 12 cm, so the stereo camera used in this paper can be modelled as a standard stereo camera model); however, we choose the left camera as the reference camera frame.
3) The IMU frame $\{I\}$: This is the frame of the IMU, with its origin at the centre of the IMU body. And the x, y, z-axes denotes the front, left, up direction of the IMU body respectively.

The relationship of the three reference frames is shown in Figure 1. The relative pose between frame {I} and frame $\{C\}$ has been carefully calibrated and is constant, while the transition between frame $\{I\}$ and frame $\{W\}$ is variable. To determine the transition of the IMU is the main purpose for our process. For this target, we consider the initial IMU position as the origin of the frame $\{W\}$.

In the following section, we denote scalars in simple italic font (*a, b, c*); and denote vectors, matrix in boldface non-italic font (**R, p**). In order to express a vector with respect to a specific reference frame, a superscript identifying the frame is appended to the vector, e.g., $\mathbf{v}^W$ for the vector $\mathbf{v}$ expressed in the frame $\{W\}$. If a vector or matrix describes the relative motion, we combine subscript letters to designate the frames, e.g. $\mathbf{p}_{IW}$ and $\mathbf{R}_{IW}$ represent the translation vector and rotation matrix from the frame $\{W\}$ to the frame $\{I\}$, respectively.

Furthermore, we utilise both the identity matrix $\mathbf{I}$ and the zero matrix $\mathbf{0}$ frequently. We use subscripts to indicate the sizes of these matrices, e.g. $\mathbf{I}_3$ represents the 3×3 identity matrix, and $\mathbf{0}_{3\times6}$ represents the 3×6 zero matrix. The vector (or matrix) transpose is identified by a superscript T, as in $\mathbf{x}^T$ (or $\mathbf{R}^T$). As for variable x and its variants $\tilde{x}, \hat{x}$, they indicate the real quantity, predicted quantity and estimated quantity of the variable respectively.

3.3. Feature Points Parameterisation. We use Cartesian 3D points to represent near feature, and Inverse Depth pointed to represent the far feature. In this paper, we refer to the Cartesian 3D and the Inverse Depth simply as 3D and ID respectively. And the standard representation for feature point in terms of a 3D point is

$$\mathbf{p}_{3D} = [X_{3D}, Y_{3D}, Z_{3D}]^T \tag{1}$$

while the ID point is

$$\mathbf{p}_{ID} = [X_C, Y_C, Z_C, \psi, \phi, \rho]^T \tag{2}$$

An ID vector can be converted into a 3D vector by the following transition

$$\mathbf{p}_{3D} = \mathbf{p}_{CW}^W + \frac{1}{\rho}\mathbf{m}(\psi, \phi) = [X_C, Y_C, Z_C]^T + \frac{1}{\rho}\mathbf{m}(\psi, \phi) \tag{3}$$

Where $\mathbf{p}_{CW}^W$ is the first camera position from which the feature was first observed, ρ is the inverse of the feature depth, and $\mathbf{m}$ is the direction of the ray passing through the image point which can be denoted by ψ, φ azimuth and elevation

$$\mathbf{m}(\psi, \phi) = [\cos\phi\cos\psi, \cos\phi\sin\psi, \sin\phi]^T \tag{4}$$

4. SYSTEM MODELLING

4.1. States Representation. We integrate the stereo images and MIMU measurements based on the IEKF estimator. The state vector used in this paper consists of IMU-related state vector and feature-related state vector, that is

$$\mathbf{x}(t) = \left[\mathbf{x}_I^T(t), \mathbf{x}_F^T(t)\right]^T \tag{5}$$

where $\mathbf{x}(t)$ is the complete state vector, $\mathbf{x}_I(t)$ is the IMU sensor-related state vector, and $\mathbf{x}_F(t)$ is the feature-related state vector.

The purpose of this process is to determine the position, velocity, attitude of the IMU. Additionally, we take the biases of the inertial sensors into consideration. Our IMU sensor-related state 16×1 vector is

$$\mathbf{x}_I(t) = \left[\mathbf{q}_{WI}^T(t), \left(\mathbf{v}_{IW}^W(t)\right)^T, \left(\mathbf{p}_{IW}^W(t)\right)^T, \mathbf{b}_g^T(t), \mathbf{b}_a^T(t)\right]^T \tag{6}$$

where $\mathbf{q}_{WI}(t)$ is a 4×1 vector which denotes a rotation quaternion (Diebel, 2006) from the IMU frame $\{I\}$ to the world frame $\{W\}$, $\mathbf{v}_{IW}^{W}(t)$ denotes the linear velocity of the IMU with respect to the world frame that expressed in the world frame, $\mathbf{p}_{IW}^{W}(t)$ is the position of IMU in the world frame, $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ are the IMU gyroscope and accelerometer biases, respectively.

In the case of an unknown environment, the positions of the detected features are unknown before. The true feature depths relative to the stereo cameras vary from near to far. We utilise the ID point to denote the features at far or infinity, and 3D point for near features. Then, the features-related state vector is

$$\mathbf{x}_F(t) = \left[\mathbf{x}_{3D}^T(t), \mathbf{x}_{ID}^T(t)\right]^T \tag{7}$$

where

$$\mathbf{x}_{3D}(t) = \left[\mathbf{p}_{1,3D}^T(t), \mathbf{p}_{2,3D}^T(t), ..., \mathbf{p}_{M,3D}^T(t)\right]^T \tag{8}$$

$$\mathbf{x}_{ID}(t) = \left[\mathbf{p}_{M+1,ID}^T(t), \mathbf{p}_{M+2,ID}^T(t), ..., \mathbf{p}_{M+N,ID}^T(t)\right]^T \tag{9}$$

Both $\mathbf{x}_{3D}(t)$ and $\mathbf{x}_{ID}(t)$ are expressed in the world frame. The number of 3D points and ID points are M, N respectively. It should be noted that both M and N are variable due to the different image textures coming from changeable scenes.

4.2. *Process Model.* The system model describes the time evolution of the IMU-related state $\mathbf{x}_I(t)$ and the features-related state $\mathbf{x}_F(t)$. In our approach, the biases of inertial sensor $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ are modelled as random walk processes, driven by zero-mean white Gaussian noise vectors, $\mathbf{n}_g$ and $\mathbf{n}_a$, with covariance matrices $\mathbf{Q}_g$ and $\mathbf{Q}_a$ respectively. As for the rotational angular velocity of the earth, it is too small, almost drowned in the noise of the MIMU gyroscope, so we do not take it into consideration. The system model of the IMU is given by the following equations (Titterton and Weston, 2004):

$$\dot{\mathbf{q}}_{WI}(t) = 0.5\boldsymbol{\Omega}\big(\boldsymbol{\omega}_m(t) - \mathbf{b}_g(t)\big)\mathbf{q}_{WI}(t) \tag{10}$$

$$\dot{\mathbf{v}}_{IW}^{W}(t) = \mathbf{R}(\mathbf{q}_{WI}(t))(\mathbf{f}_m(t) - \mathbf{b}_a(t)) + \mathbf{g}^W \tag{11}$$

$$\dot{\mathbf{p}}_{IW}^{W}(t) = \mathbf{v}_I^W(t) \tag{12}$$

$$\dot{\mathbf{b}}_g(t) = \mathbf{n}_g(t),\ \dot{\mathbf{b}}_a(t) = \mathbf{n}_a(t) \tag{13}$$

where $\boldsymbol{\omega}_m(t)$ and $\mathbf{f}_m(t)$ are the angular velocity and the specific force measured by the IMU. $\mathbf{g}^W$ is the local gravity vector in the world frame. $\mathbf{R}(\mathbf{q})$ is the rotational matrix corresponding to a quaternion vector. $\boldsymbol{\omega} = \boldsymbol{\omega}_m(t) - \mathbf{b}_g(t) = [\omega_x, \omega_y, \omega_z]^T$ is the rotational velocity of the IMU expressed in the IMU frame, and

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -\lfloor\boldsymbol{\omega}\times\rfloor \end{bmatrix},\ \lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{14}$$

For the feature-related state, we consider the static features in this approach, and all the features are expressed in the world frame, then we have

$$\dot{\mathbf{x}}_F(t) = \mathbf{0} \tag{15}$$

4.3. *Measurement Model.* In order to make full use of the potential properties of the stereo camera and MIMU, the navigation and features tracking algorithms are tightly-coupled. We consider the feature's pixel coordinates in the image plane as the measurement. The camera used in our approach is a perspective camera, so an ideal pinhole projective model for our camera is feasible. Additionally, the stereo camera gives a rectified image pair and the intrinsic parameters of the camera were known before. However, if the camera intrinsic and distortion parameters were unknown, we can obtain those parameters by observing a chequered target and performing the camera calibration with Bouguet's camera calibration toolbox (Bouguet 2006), although this is not in the scope of the work presented here.

Measurement $\mathbf{z}_i$ is the projection of the ith feature, at position $\mathbf{p}_{fi}^C = [x_i, y_i, z_i]^T$ in the camera frame onto the image plane, and the projective camera measurement model is:

$$\mathbf{z}_i(t) = \begin{bmatrix} u_i \\ v_i \end{bmatrix} + \boldsymbol{\eta}_i = \begin{bmatrix} fx_i/z_i + u_0 \\ fy_i/z_i + v_0 \end{bmatrix} + \boldsymbol{\eta}_i \tag{16}$$

where u_0, v_0 is the camera principal point, f is the focal length, and $\boldsymbol{\eta}_i$ is the measurement 2×1 noise vector with covariance matrix $\mathbf{R}_i = \sigma_i^2 \mathbf{I}_2$.

To estimate the position of an observed feature in the camera frame, the methods depend on the type of feature. For features in 3D point

$$\begin{aligned} \mathbf{p}_{fi}^C &= \mathbf{h}_{3D}\left(\mathbf{x}_I(t), \mathbf{p}_{i,3D}(t)\right) \\ &= \mathbf{R}_{CI}\left(\mathbf{R}_{WI}^T(\mathbf{q}_{WI}(t))\left(\mathbf{p}_{i,3D}(t) - \mathbf{p}_{IW}^W(t)\right) - \mathbf{p}_{CI}^I\right) \end{aligned} \tag{17}$$

and for features in ID point

$$\begin{aligned} \mathbf{p}_{fi}^C &= \mathbf{h}_{ID}\left(\mathbf{x}_I(t), \mathbf{p}_{i,ID}(t)\right) \\ &= \mathbf{R}_{CI}\left(\mathbf{R}_{WI}^T\left(\mathbf{p}_{CW,i}^W + \frac{1}{\rho_i}\mathbf{m}(\theta_i, \phi_i) - \mathbf{p}_{IW}^W\right) - \mathbf{p}_{CI}^I\right) \end{aligned} \tag{18}$$

where the sub-index i denotes the ith feature; $\mathbf{R}_{CI}$, $\mathbf{p}_{CI}^I$ are the relative rotation matrix and translation between IMU frame and the camera frame, which are known in advance.

5. THE IEKF-BASED INTEGRATION ALGORITHM

5.1. *Linearization Error Model.* We wish to write the error-state equations of the system model. For brevity, we do not indicate dependence on time in the following section. The IMU-related error-state is defined as

$$\delta\mathbf{x}_I = \left[\delta\boldsymbol{\theta}_{WI}^T, \left(\delta\mathbf{v}_{IW}^W\right)^T, \left(\delta\mathbf{p}_{IW}^W\right)^T, \delta\mathbf{b}_g^T, \delta\mathbf{b}_a^T\right]^T \tag{19}$$

For the IMU position, velocity and biases, the error is defined as $\delta\mathbf{x} = \mathbf{x} - \tilde{\mathbf{x}}$, where $\mathbf{x}$ is a true quantity, and $\tilde{\mathbf{x}}$ is the estimate of the quantity. However, for a quaternion, if the true quaternion is denoted as $\mathbf{q}$ and the estimate $\tilde{\mathbf{q}}$, a different error definition will be employed:

$$\mathbf{q} = \tilde{\mathbf{q}} \otimes \delta\mathbf{q} = \tilde{\mathbf{q}} \otimes \left[1, \delta\boldsymbol{\theta}^T/2\right]^T \tag{20}$$

where the operator of $\otimes$ denotes quaternion multiplication (Titterton and Weston, 2004). It is worthwhile to note that the attitude error $\delta\boldsymbol{\theta}$ is a 3×1 vector while a quaternion $\mathbf{q}$ is a 4×1 vector. Therefore, the dimension of the vector $\delta\mathbf{x}_I$ is 15 which is a little different from that of the vector $\mathbf{x}_I$.

Similarly, the feature-related state vector is defined as:

$$\delta\mathbf{x}_F = \begin{bmatrix} \delta\mathbf{x}_{3D} \\ \delta\mathbf{x}_{ID} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{3D} - \tilde{\mathbf{x}}_{3D} \\ \mathbf{x}_{ID} - \tilde{\mathbf{x}}_{ID} \end{bmatrix} \tag{21}$$

Note that the number of the 3D and ID point are M and N respectively, so the dimension of the vector $\delta\mathbf{x}_F$ is $3M + 6N$. The complete error-state is defined as:

$$\delta\mathbf{x} = \left[\delta\mathbf{x}_I^T, \delta\mathbf{x}_F^T\right]^T = \left[\delta\mathbf{x}_I^T, \delta\mathbf{x}_{3D}^T, \delta\mathbf{x}_{ID}^T\right]^T \tag{22}$$

then we obtain

$$\begin{bmatrix} \delta\dot{\mathbf{x}}_I \\ \delta\dot{\mathbf{x}}_F \end{bmatrix} = \begin{bmatrix} \mathbf{F}_I & \mathbf{0}_{15\times(3M+6N)} \\ \mathbf{0}_{(3M+6N)\times 15} & \mathbf{0}_{(3M+6N)\times(3M+6N)} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_I \\ \delta\mathbf{x}_F \end{bmatrix} + \begin{bmatrix} \mathbf{n}_I \\ \mathbf{0}_{(3M+6N)\times 1} \end{bmatrix} \tag{23}$$

where

$$\mathbf{F}_I = \begin{bmatrix} -\lfloor(\boldsymbol{\omega}_m - \mathbf{b}_g)\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ -\mathbf{R}_{WI}\lfloor(\mathbf{a}_m - \mathbf{b}_a)\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{R}_{WI} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} \end{bmatrix}, \mathbf{n}_I = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{n}_g \\ \mathbf{n}_a \end{bmatrix} \tag{24}$$

As shown in Equations (7) to (9), the total number of features observed in one image is $M + N$. We stack all the individual measurements to form one $2(M + N) \times 1$ measurement vector

$$\mathbf{z} = \left[\mathbf{z}_{1,3D}^T, ..., \mathbf{z}_{M,3D}^T, \mathbf{z}_{M+1,ID}^T, ..., \mathbf{z}_{M+N,ID}^T\right]^T \tag{25}$$

and the error measurement model is

$$\delta\mathbf{z} = \mathbf{z} - \tilde{\mathbf{z}} = \mathbf{H}\delta\mathbf{x} + \boldsymbol{\eta} \tag{26}$$

Where $\boldsymbol{\eta} = \left[\boldsymbol{\eta}_1^T, ..., \boldsymbol{\eta}_{M+N}^T\right]^T$ is the measurement noise vector with the covariance matrix $\mathbf{R} = blkdiag(\mathbf{R}_1, \ldots, \mathbf{R}_{M+N})$;

$$\mathbf{H} = \left[\mathbf{H}_{1,3D}^T, ..., \mathbf{H}_{M,3D}^T, \mathbf{H}_{M+1,ID}^T, ..., \mathbf{H}_{M+N,ID}^T\right]^T \tag{27}$$

is the measurement Jacobian matrix. The individual measurement matrix $\mathbf{H}_i$ is computed as follows:

$$\begin{aligned} \mathbf{H}_{i,3D} &= \mathbf{J}_{i,z}\mathbf{R}_{CI}\left[\ \left\lfloor\left(\mathbf{R}_{WI}^T\left(\mathbf{p}_{fi}^W - \mathbf{p}_{IW}^W\right)\right)\times\right\rfloor \quad \mathbf{0}_{3\times3} \quad -\mathbf{R}_{WI}^T \quad \mathbf{0}_{6\times3} \quad \ldots \quad \mathbf{R}_{WI}^T \quad \ldots\ \right] \\ \mathbf{H}_{i,ID} &= \mathbf{J}_{i,z}\mathbf{R}_{CI}\left[\ \left\lfloor\left(\mathbf{R}_{WI}^T\left(\mathbf{p}_{fi}^W - \mathbf{p}_{IW}^W\right)\right)\times\right\rfloor \quad \mathbf{0}_{3\times3} \quad -\mathbf{R}_{WI}^T \quad \mathbf{0}_{6\times3} \quad \ldots \quad \mathbf{J}_{i,ID} \quad \ldots\ \right] \end{aligned} \tag{28}$$

with

$$\begin{aligned}
\mathbf{J}_{i,z} &= \frac{f}{\tilde{z}_i^2}\begin{bmatrix} \tilde{z}_i & 0 & -\tilde{x}_i \\ 0 & \tilde{z}_i & -\tilde{y}_i \end{bmatrix}, \\
\mathbf{J}_{i,ID} &= \mathbf{R}_{WI}^T\begin{bmatrix} \mathbf{I}_3 & [\mathbf{J}_{i,\mathbf{m}}, \mathbf{J}_{i,\rho}] \end{bmatrix}, \\
\mathbf{J}_{i,\mathbf{m}} &= \frac{1}{\rho_i}\begin{bmatrix} -\cos\phi_i \sin\psi_i & -\sin\phi_i \cos\psi_i \\ \cos\phi_i \cos\psi_i & -\sin\phi_i \sin\psi_i \\ 0 & \cos\phi_i \end{bmatrix}, \\
\mathbf{J}_{i,\rho} &= \frac{1}{\rho_i}\left[\left(\mathbf{p}_{CW,i}^W - \mathbf{p}_{IW}^W\right) - \mathbf{R}_{WI}\mathbf{p}_{CI}^I\right]
\end{aligned} \tag{29}$$

5.2. *Algorithm Implementation.* As shown in the previous section, the equations of the process and measurement model are nonlinear. In order to reduce the linearization error of the nonlinear equations, the IEKF scheme is employed. The IEKF fuses the inertial measurements and visual stereo image pairs in a tightly-coupled approach, as shown in Figure 2.

There are mainly four parts in the flowchart shown in Figure 2. The measurements of MIMU $\boldsymbol{\omega}_m(t)$ and $\mathbf{a}_m(t)$ are used for updating the inertial related state vector $\mathbf{x}_I(t)$ by employing Equations (10) to (13), which is the process of inertial navigation. Before inertial navigation is available, there is an alignment of the IMU using a short period of static inertial data for level alignment and magnetometer data for azimuth alignment, which is detailed in the next section.

As for the image processing part, the images coming from the stereo camera are used for tracking the observed features, and for detecting new features. During the feature tracking process, a Kai-Square test based outlier rejection method is employed by utilising predicted feature locations $\tilde{z}(t)$ and variances $S(t)$. Details on this part are also discussed in a later section.

The system states management section is an intermediate link with three main tasks: firstly, to manage the feature-related states according to the results of the image processing; secondly, to compensate the predicted states based on the estimated state errors from IEKF and finally to give the best states estimation.

The IEKF part is the core of the flowchart, with the aim of fusing the inertial measurements and stereo image sequences. The detailed process of the IEKF can be seen in the following section.

5.3. *The Pseudo Code of the IEKF Algorithm.* In order to minimize linearization errors of the nonlinear model, we employ the IEKF to update the states. The pseudo code of the algorithm is as follows:

For j = 1: IterNum

1) Employ Equations (16), (17) and (18) to predict the measurement vector $\tilde{\mathbf{z}}^j$ which is a function of the current iteration $\tilde{\mathbf{x}}_{k+1,k+1}^j$;
2) Compute the measurement Jacobian matrix $\mathbf{H}^j$ around the current iteration using Equations (27), (28) and (29);

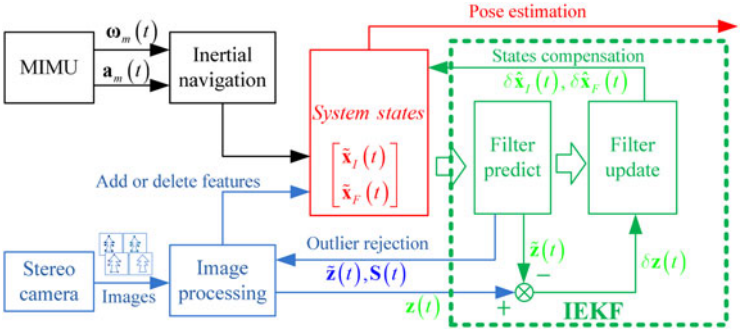


Figure 2. The flowchart of the tight integration system of the stereo camera and the MIMU. The flowchart is divided into four parts with different colours, namely inertial navigation part (black part in the flowchart); image processing (blue) consisting of feature detecting, tracking and outlier rejection; the part of IEKF (green) and the system states management part (red).

3) Compute the measurement error vector $\delta\mathbf{z}^j = \mathbf{z} - \tilde{\mathbf{z}}^j$ and its variance matrix

$$\mathbf{S}^j = \mathbf{H}^j\mathbf{P}_{k+1,k}\mathbf{H}^{j^T} + \mathbf{R} \tag{30}$$

4) Compute the Kalman gain matrix $\mathbf{K}^j = \mathbf{P}_{k+1,k}\mathbf{H}^{j^T}\left(\mathbf{S}^j\right)^{-1}$ and compute the state error correction vector $\delta\hat{\mathbf{x}}^j_{k+1,k+1} = \delta\mathbf{x}^j_{k+1,k+1} - \mathbf{K}^j\left(\mathbf{H}^j\delta\mathbf{x}^j_{k+1,k+1} - \delta\mathbf{z}^j\right)$;
5) Compensate current state $\tilde{\mathbf{x}}^{j+1}_{k+1,k+1} = \tilde{\mathbf{x}}^j_{k+1,k+1} + \delta\hat{\mathbf{x}}^j_{k+1,k+1}$, then reset the error state $\delta\mathbf{x}^{j+1}_{k+1,k+1} = \mathbf{0}$.

End

The covariance matrix of the state $\mathbf{x}$ for the final state is updated by $\mathbf{P}_{k+1,k+1} = (\mathbf{I} - \mathbf{KH})\mathbf{P}_{k+1,k}$, where the matrix $\mathbf{K}$ and $\mathbf{H}$ are from the final iteration.

5.4. *The Static Alignment of the Low-Cost IMU.* The purpose of the alignment of the low-cost IMU is to estimate an initial attitude of the IMU with respect to the world frame and the initial inertial sensor biases.

Under the static condition, a conventional method to obtain the IMU attitude is parse alignment (Titterton and Weston, 2004) by using gravity and the earth's rotation angle rate. However, it is unavailable in this work because of the high level noise of the gyroscope in the IMU which can submerge the earth's rotation angle rate. In our approach, we use gravity and accelerometer measurements to determine the tilt angle (roll and pitch), and obtain the azimuth angle using the magnetometer. The tilt angle can be calculated as follows:

$$\begin{aligned} \theta_x &= \operatorname{atan2}\left(\mathrm{g_y}/\mathrm{g_z}\right) \\ \theta_y &= -\operatorname{asin}\left(\mathrm{g_x}/\mathrm{g}\right) \end{aligned} \tag{31}$$

where θ_x, θ_y are the tilt angle roll and pitch respectively, g_x, g_y, g_z are the accelerometer measurements of the IMU, and g is the determinant of local gravity. With the knowledge of roll and pitch angles and the magnetometer measurements, we can obtain the

yaw angle with respect to the magnetic north as follows

$$\begin{bmatrix} M_{xH} \\ M_{yH} \end{bmatrix} = \begin{bmatrix} \cos\theta_y & \cos\theta_y \sin\theta_x & -\cos\theta_x \sin\theta_y \\ 0 & \cos\theta_x & \sin\theta_x \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{32}$$

$$\theta_z = \operatorname{atan} 2\left(M_{yH}, M_{xH}\right) + \delta \tag{33}$$

where M_x, M_y, M_z are the magnetometer measurements of the IMU, M_{xH}, M_{yH} are the magnetometer measurements of the IMU frame projected on the horizontal plane, and the local declination angle δ which can be determined from a lookup table based on the geographic location is added to correct for true north.

Once we get an initial attitude of the IMU, we perform a standard Kalman filter to estimate the biases of inertial sensor and refine the other IMU-related states. In this progress, the state of Kalman Filter is same to the IMU-related state in Equation (6). As for the measurement model of the filter, the position and velocity of the IMU are chosen as the measurment of the filter because of the position remaining unchanged and the velocity remaining zero during the static period. The measurement model is presented as follows:

$$\delta \mathbf{y} = \begin{bmatrix} \mathbf{0}_{3\times1} - \tilde{\mathbf{v}}_{IW}^{W} \\ \mathbf{0}_{3\times1} - \tilde{\mathbf{p}}_{IW}^{W} \end{bmatrix} = \mathbf{H}_{\mathbf{y}} \delta \mathbf{x}_I + \boldsymbol{\eta}_{\mathbf{y}} \tag{34}$$

where

$$\mathbf{H}_{\mathbf{y}} = [\, \mathbf{0}_{3\times3} \quad \mathbf{I}_3 \quad \mathbf{I}_3 \quad \mathbf{0}_{3\times6} \,] \tag{35}$$

5.5. *Feature Detection, Tracking, and Outlier Rejection.* In order to find salient features in the images efficiently, we use the fast corner detection (FAST) algorithm proposed by Rosten and Drummond (2005; 2006) to detect corner features in the left image. The major advantage of the FAST algorithm is that it can reach an accurate corner localization in an image with high efficiency. Once new features are detected, then we track them in the current right image as well as the next left images by employing the Kanade Lucas Tomasi (KLT) tracker (Shi and Tomasi, 1994). The KLT tracker allows tracking features over long image sequences and undergoing larger changes by applying an affine-distortion model to each feature. Additionally, the left-right image matching and the current-next image matching play different roles in our approach, they are used for features initialization and filter updating respectively.

However, matched points are usually contaminated by outliers which may be caused by image noise, occlusion, image blur and changes in viewpoint. Attention has been paid to outlier rejection. For left-right matching, we employ the epipolar geometry constraint for outlier removal. Since the stereo cameras used in our work are aligned with each other, the epipolar geometry constraint (Szeliski, 2011) can be simplified as $v_R - v_L = 0 \pm S$. In this formulation, v_R, v_L are the pixel vertical coordinates in the right and left image, and S (in our case, the value of S is 1·5 pixel) is a threshold previously defined for acceptable noise level.

In the case of current and next image matching, we employ a Chi-square test to detect and reject the outliers. At every epoch, when a new measurement is available, then we have

$$\delta \mathbf{z}_i^T \mathbf{S}_i^{-1} \delta \mathbf{z}_i \sim \chi^2(2) \tag{36}$$

Where $\delta \mathbf{z}_i$ is a 2×1 measurement residual vector with its variance $\mathbf{S}_i$ which can obtain from Equation (30), and $\chi^2(2)$ represents a Chi-square distribution with a degree of 2. We reject any feature measurement whose residual is above the threshold.

5.6. *Features Initialization and Management.* Once a new feature is detected, it is preprocessed with a feature initialization procedure. Firstly, features are classified according to the comparison between their disparity and a given threshold (e.g. 7 pixels). Features with high disparity are initialized as 3D features, and others are initialized as ID features. For 3D feature, the initialization procedure is as follows:

$$\mathbf{p}_{3D} = f\left(\mathbf{q}_{WI}, \mathbf{p}_{IW}^{W}, \mathbf{p}_{f}^{C}\right) = \mathbf{R}_{WI}\mathbf{R}_{CI}^{T}\mathbf{p}_{f}^{C} + \mathbf{R}_{WI}\mathbf{p}_{CI}^{I} + \mathbf{p}_{IW}^{W} \tag{37}$$

$$\mathbf{p}_{f}^{C} = g(u_L, u_R, v_L, v_R) = \left[\frac{b(u_L - u_0)}{d}, \frac{b(v_L - v_0)}{d}, \frac{bf}{d}\right]^T \tag{38}$$

where (u, v) is the pixel coordinate in the image, b is the baseline of the stereo camera, and $d = u_L - u_R$ is the disparity.

For the ID feature, we have

$$\begin{aligned} \mathbf{p}_{ID} &= h\left(\mathbf{q}_{WI}, \mathbf{p}_{IW}^{W}, u_L, u_R, v_L, v_R\right) \\ [X_C, Y_C, Z_C]^T &= \mathbf{p}_{IW}^{W} + \mathbf{R}_{WI}\mathbf{p}_{CI}^{I} \\ \psi &= \arctan\left(n_y, n_x\right) \\ \phi &= \arctan\left(n_z, \sqrt{{n_x}^2 + {n_y}^2}\right) \\ \rho &= d/bf \end{aligned} \tag{39}$$

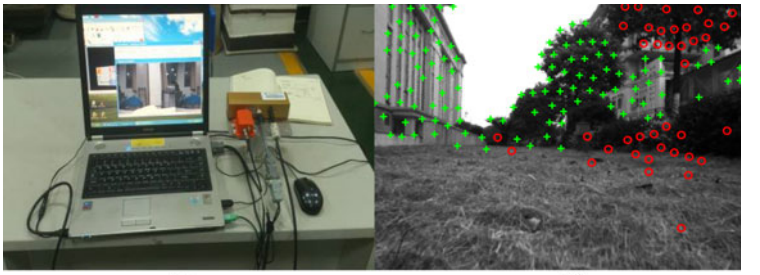
where $[n_x, n_y, n_z]^T = \mathbf{R}_{WI}\mathbf{R}_{CI}^{T}[u - u_0, v - v_0, f]^T$. Note that the covariance of $\mathbf{p}_{3D}$, $\mathbf{p}_{ID}$ can be derived from the image measurement error covariance matrix $\mathbf{R}_i$ and state covariance matrix $\mathbf{P}$.

In order to reduce the number of the state degrees of freedom, we convert the ID point to the 3D point properly. The analysis of the linearity of the functions that model both depth point and ID point distributions needs to be taken into consideration. We utilise a linearity index presented by Civera et al. (2008) to get around this issue. We calculate the linearity index at each step, and compare with a linearity threshold to determine whether covert or not.

6. EXPERIMENT AND RESULTS

6.1. *Experimental Procedure.* In order to evaluate the performance of the proposed algorithm, we performed experiments using a test rig which consists of the VIS and a laptop for data acquisition (Figure 3(a)). The cameras' field of view is 70° with a focal length of 3.8 mm, and the resolution of image is 640×480 pixels. Stereo images are sampled at a rate of 10 Hz whereas MIMU provides measurements at 100 Hz. In the following section, we will present a typical result from an outdoor test which is a representative sample of the performance of the proposed algorithm across a series of trials.

At the beginning of the experiment, we put the sensor on the ground to initialise the attitude and inertial sensor biases with a stationary alignment for approximately one minute. After this alignment period, the sensor was picked up and moved in the



(a) (b)

Figure 3. (a) Data collection system, which consists of a laptop and the visual-inertial system. (b) A sample image from outdoor data collection, which contains both near (circles) and far (crosses) features.

hand, at a walking speed of 3–5 km/h. The profile is a closed rectangle loop, with a total length of about 120 m, and a height of about 1·5 m above the ground during moving. In the scene, there are both near and far features, as shown in Figure 3(b), which is a sample image from the data collection. Image sequence and corresponding MIMU data were collected in a laptop. We processed the data in MATLAB with the proposed algorithm on a Core 2 Duo 2.5 GHz desktop computer.

6.2. *Results*

6.2.1. *Inertial-only Solution and Results.* The collected data are composed of 1610 stereo pairs and 21800 inertial measurements. We did a stationary alignment by using the first 6100 inertial measurements. The main purpose of the alignment procedure is to obtain an intial attitude of the IMU with respect to the world frame (in our case, the x, y, z-axes of the world frame are aligned with north, west and up direction respectively) and to estimate the inertial sensor's biases. Since the static state of the IMU, the estimated velocity shown in Figure 5 and position shown in Figure 6 are close to zero in the first 61 seconds. In order to show the importance of estimating and compensating sensor biases of the MIMU, for inertial-only solution, we performed two separate inertial navigation experiments. The two inertial navigation processes use the same intial position, velocity and attitude, but are different in whether compensating the sensor biases or not. We use "iner-only-uncomp" to denote the inertial-only solution without compensating sensor biases and "iner-only-comp" to denote that with compensating biases. The results are compared in Figures 4 to 6. Due to the accumulative error of the biases and noises of the low-cost inertial sensor, the iner-only-uncomp results deteriorated tremendously making them barely useful for navigation tasks alone. On the other hand, the iner-only-comp approach, which compensated the sensor biases, is able to keep an acceptable error for a short time. The comparison evaluates the importance of estimating and compensating the inertial sensor biases for the navigation task, especially the low-cost inertial sensors. In addition, it is necessary to point out that in spite of the considerable accumulative error in the iner-only solution, it is enough for VIS, because the consecutive measured image can revise the accumulative error periodically.

6.2.2. *Comparison of Iner-only, Vis-only and Vis-iner Solution.* For simplicity, *iner-only, vis-only*, and *vis-iner* are used to represent three different solutions, namely inertial-only solution, visual-only solution and inertial-visual solution respectively. The iner-only solution has been discussed above. As for the vis-only solution, we

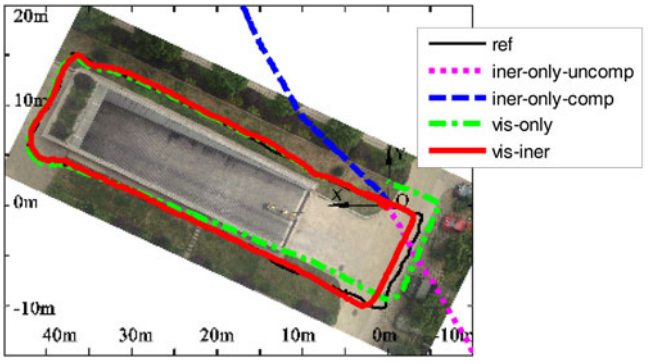


Figure 4. Estimated path in the horizontal plane for iner-only-uncomp (sensor biases ignored), iner-only-comp (sensor biases compensated), vis-only and vis-iner solution. The iner-only-uncomp and iner-only-comp solutions exceed the scale of the map after 70 and 76 seconds, respectively. The path estimated by vis-iner agrees well with the known path (the black line) and shows smaller errors in position and heading than the vis-only solution.

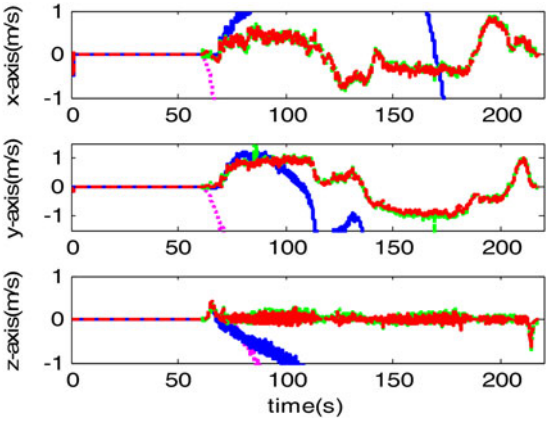


Figure 5. Estimated velocity comparison for iner-only-uncomp (magenta dotted line), iner-only-comp (blue solid line), vis-only (green dash-dotted line) and vis-iner (red dash line) solution. After the 61 seconds alignment period, the estimated velocity by iner-only-uncomp diverged tremendously while the iner-only-comp keeps stable for a short time. The vis-only and vis-iner solutions have similar results. In spite of that, the vis-iner shows a smoother solution than the vis-only.

employed the algorithm presented in Civera et al. (2010). The estimated trajectories for vis-only and vis-iner are also overlaid on the horizontal plane in Figure 4. The black line in Figure 4 is the reference path which is obtained by manually tracking the real trajectory according to the image sequences in the test scenario and overlaying the tracked trajectory on the horizontal plane. As shown in Figure 4, both the vis-only and vis-iner estimated trajectory generally correspond to the real path, which improves the position error by several orders of magnitude over the iner-only result. Detailed velocity and position estimation are compared in Figures 5 and 6.

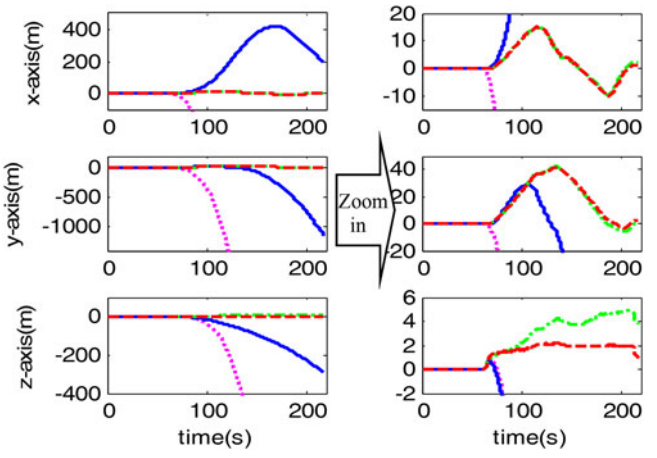


Figure 6. Estimated position comparison for iner-only-uncomp (magenta dotted line), iner-only-comp (blue solid line), vis-only (green dash-dotted line) and vis-iner (red dash line) solution. The left column shows that the vis-iner result has significant improvement over the iner-only results. The zoomed in position results are shown in the right column. Compared to the vis-only result, the vis-iner has higher estimation precision, especially in the height direction (z-axis).

Note that, in this case, the reason for the similar results of the vis-only and vis-iner solution is that there are numerous features in the scene. The profiles of vis-iner and vis-only result (as shown in Figure 4) are more or less the same, however, the vis-iner result shows less closure error than the vis-only result due to the smaller heading error in the vis-iner solution. As for the height estimation, as shown in the right and bottom box of Figure 6, the height estimated by the vis-iner solution is about 1.5 m and remains stable, which is closer to the actual height. The total length of the trajectory is about 120 m, and the position error of the vis-iner solution is less than 3 m.

6.2.3. *Comparison of Using Different Types of Features.* In order to compare the performances of utilising different features, we ran our algorithm three times with near features, far features and both near and far features (designated as "N&F") respectively. The results are compared in Figure 7. The results show that the far features-only solution has a good heading estimation, but a poor position estimation. Meanwhile, the near feature-only solution shows the opposite performance. The results prove that the near features have much more range information whereas the far features provide more bearing information. However, in our approach, the N&F solution takes advantage of both the near and far features, providing a better performance than either of them.

6.2.4. *Feature Tracking and Management.* In our approach, IEKF is used for the integration of the visual and inertial data, which can estimate the states' value as well as their variance. Figure 8 shows the features in the image plane and the world frame. The results show that a large number of features-matching succeeded while a few features failed. However, the failed features are mainly the features running out of the field of view or shaking features. It is worth noting that this is the result of stable filter running, in which the variance of the feature is estimated correctly, and can be used for outlier rejection.

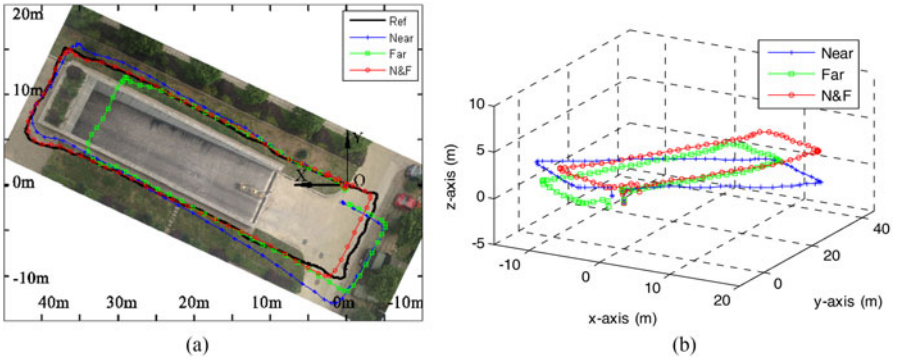


Figure 7. Performance comparison for using near, far, and N&F features. (a) shows the estimated path in the horizontal plane and (b) gives the 3D trajectory estimation.

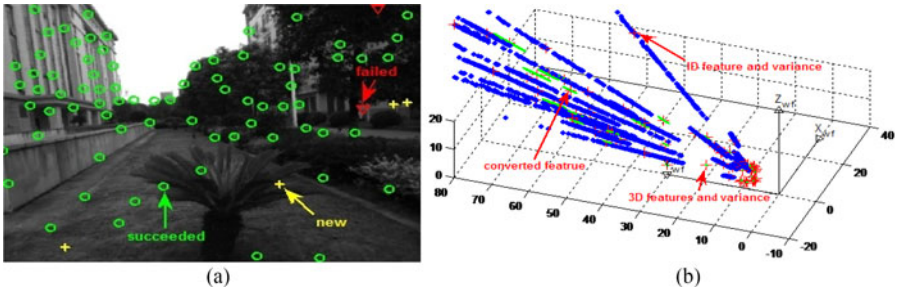


Figure 8. The features in the image (a) and their position and variance estimation in the world frame (b). Most features in the image (a) matched successfully, with a few features failing and being newly detected. In the 3D image (b), the red crosses indicate the position estimation of currently tracked features, the blue and green ellipse indicate the variance of the features. Note that, the blue and green ellipse look like bold lines radiating from a common centre (in our case, the centre is the camera), which is the result of the great uncertainties in the range direction.

We abandoned the failed features and detected a new one, the features in the world frame are classified into ID and 3D representation (as shown in Figure 8 (b)). For the ID features, they have great uncertainties in the range when they are initialised. However, as the camera moves, the baseline increases, reducing the uncertainty in the range of an ID feature. Once the depth estimate of the ID feature is sufficiently accurate, we convert the 6D vector to the 3D representation, which will reduce the computational burden. The dynamic process can be seen in an accompanying video file available on request from the corresponding author.

6.2.5. *Running Time of The System.* In order to evaluate the real time performance of our system, the execution time of data processing is presented. At the beginning, the algorithm just does the static alignment which does not utilise the images, so the execution speed is very fast, as shown in Figure 9 (a). After the static alignment, the system begins to execute the IEKF procedure which mainly contains several time-consuming steps, namely image load, feature detect, feature track, IEKF update, feature manage and others, and their detailed execution time for one second data

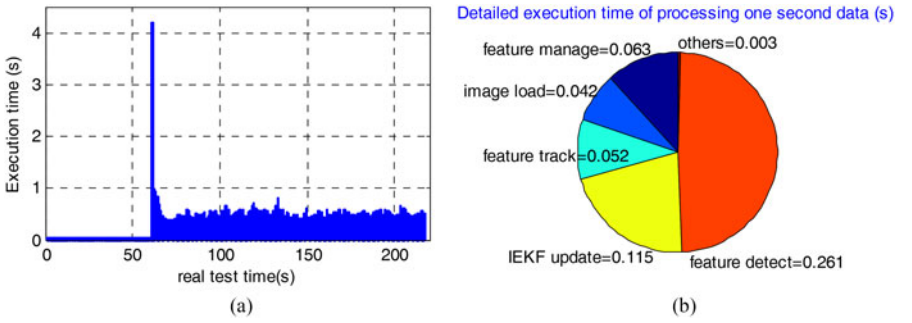


Figure 9. The execution time of processing 1 s data which consists of 100 inertial measurements and 10 image pairs. (a) shows the execution time versus the real test time and (b) shows the main steps and their execution times.

are shown in Figure 9 (b). Clearly, the feature detection consumes the highest time and is followed by the IEKF update procedure. At time 61 s, the execution time increases dramatically, as shown in Figure 9 (a). The reason is that this is the first image frame which spends much time on detecting the new features. After that, the execution time falls to an average level which is about 0.5 s. The average execution time shows that our system can work in real time and has a good real time performance.

7. CONCLUSIONS. In this paper, we combined a low-cost inertial sensor and stereo cameras for an autonomous navigation task. Both the IMU and cameras are passive sensors, which allow the vehicle to navigate in GPS denied/shaded environments. In contrast to previous approaches, we account for both near and far features in the scene, in which the near features contain distance and orientation information whereas the far features provide orientation information. The two kinds of features are represented in terms of 3D points and Inverse Depth points respectively, and fused with inertial data in IEKF. The number of the features can be inconstant, and the ID features are properly converted into 3D ones for the sake of reducing the computational burden and storage space.

The proposed algorithm has been applied to an outdoor test. The comparison shows that the vis-iner approach has more multiple orders of magnitude improvement than the inertial-only solution, and has a more precise and smooth motion estimation than the visual-only. What is more, the result also shows that using both near and far features has a certain advantage over using only one of them, with the position error being less than 3 m for an outdoor path of 120 m length. With the results seen herein, the tight integration of stereo cameras and low-cost inertial sensor proposed in the paper has a precise motion estimation performance, and can be used for autonomous navigation tasks.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No.61104201). Sincere appreciation is given to Yujie Wang for his advice.

REFERENCES

Armesto, L., Tornero, J. and Vincze, M. (2007). Fast Ego-motion Estimation with Multi-rate Fusion of Inertial and Vision. *The International Journal of Robotics Research*, **26**(6), 577–89.

Bouguet, J.-Y. (2006). Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.).

Carrillo, L.R.G., López, A.E.D., Lozano, R. and Pégard, C. (2012). Combining Stereo Vision and Inertial Navigation System for a Quad-Rotor UAV. *Journal of Intelligent Robot Systems*, **65**, 373–87.

Chowdhary, G., Johnson, E.N., Magree, D., Wu, A and Shein, A. (2013). GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *Journal of Field Robotics*, **30**(3), 415–38.

Civera, J., Davison, A.J. and Montiel, J.M.M. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, **24**(5), 932–45.

Civera, J., Grasa, O.G., Davison, A.J. and Montiel, J.M.M. (2010). 1-Point RANSAC for Extended Kalman Filtering: Application to Real-Time Structure from Motion and Visual Odemetry. *Journal of Field Robotics*, **27**(5), 609–31.

Corke, P., Lobo, J. and Dias, J. (2007). An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, **26**(6), 519–35.

Davison, A. (1998). *Mobile robot navigation using active vision*, Oxford, U.K. Oxford.

Davison, A. (2003). Real-time simultaneous localization and mapping with a single camera, in *Proceeding of the Ninth IEEE International Conference on Computer Vission*, 1403–10.

Davison, A.J. and. Murray, D.W. (2002). Simultaneous localization and mapbuilding using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(7), 865–80.

Davison, A.J., Reid, I.D, Molton, N.D. and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal.Mach. Intell.*, **29**(6), 1052–67.

Diebel, J. (2006). Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. *Matrix*, **58**, 15–16.

Feng, G., Wu, W. and Wang, J. (2012). Observability analysis of a matrix Kalman filter-based navigation system using visual /inertial /magnetic sensors. *Sensors*, **12**(7), 8877–94.

Kelly, J. and Saripalli, S. (2008). Combined Visual and Inertial Navigation for an Unmanned Aerial Vehicle. *Field and Service Robotics*, **42**, 255–64.

Kelly, J. and Sukhatme, G.S. (2011). Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration. *International Journal of Robotics Research*, **30**(1), 56–79.

Kim, J.H. and Sukkarieh, S. (2003). Airborne simultaneous localisation and map building, in *IEEE International Conference on Robotic and Automation*, 406–11.

Lang, P. and Pinz, A. (2005). Calibration of Hybrid Vision / Inertial Tracking Systems, in *Proceedings of the 2nd InerVis: Workshop on Integration of Vision and Inertial Senors*, Barcelona, Spain, 527–33.

Lobo, J. and Dias, J. (2007). Relative Pose Calibration Between Visual and Inertial Sensors. *The International Journal of Robotics Research*, **26**(6), 561–75.

Mirzaei, F.M. and Roumeliotis, S.I. (2008). A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. *IEEE Transactions on Robotics*, **24**(5), 1143–56.

Mourikis, A. and Roumeliotis, S.I. (2007). A multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation, in *IEEE Inernational Conference in Robotics and Automation,* Roma, Italy, 3565–72.

Noureldin, A., El-Shafie, A. and Bayoumi, M. (2011). GPS/INS integration utilizing dynamic neural networks for vehicular navigation. *Information Fusion*, **12**, 48–57.

Paz, L.M., Piniés, P., Tardós, J.D. and Neira, A.J.E. (2008). Large-Scale 6-DOF SLAM With Stereo-in-Hand. *IEEE Transactions on Robotics*, **24**(5), 946–57.

Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking, in *IEEE International Conference on Computer Vision*, 1508–11.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection, in *European Conference on Computer Vision*, 430–43.

Se, S., Lowe, D. and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, **21**(8), 735–58.

Shi, J. and Tomasi, C. (1994). Good Features to Track, in *IEEE Conference on Computer Vision and Pattern Recognition*, 593–600.

Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*, Springer.

Titterton, D.H. and Weston, J.L. (2004). *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers: London, UK.

Veth, M. and Raquet, J. (2007). Fusing low-cost image and inertial sensors for passive navigation. *Journal of the Institute of Navigation*, **54**(1), 11–20.

Yun, S., Lee, Y.J. and Sung, S. (2013). IMU/Vision/Lidar Integrated Navigation System in GNSS Denied Environments, in *2013 IEEE Aerospace Conference* (IEEE Aerospace Conference Proceedings).