2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A Sliding-Window Visual-IMU Odometer Based on Tri-focal Tensor Geometry

Jwu-Sheng Hu and Ming-Yuan Chen

***Abstract*—This paper presents an odometer architecture which combines a monocular camera and an inertial measurement unit (IMU). The trifocal tensor geometry relationship between three images is used as camera measurement information, which makes the proposed method without estimating the 3D position of feature point. In other words, the proposed method does not have to reconstruct environment. Meanwhile, the camera pose corresponding to each of the three images are refined in filter to form a multi-state constraint Kalman filter (MSCKF). Consequently, this paper proposes a sliding window odometry which has a balance between computational cost and accuracy. Compared with traditional visual odometry or simultaneous localization and mapping (SLAM) method, the proposed method not only meets the requirement of odometer in the ego-motion estimation, but also suit for real-time application. This paper further proposes a random sample consensus (RANSAC) algorithm which is based on three views geometry. The RANSAC algorithm can effectively reject feature points which are mismatch or located on independently moving objects, thus it make the overall algorithm capable of operating in dynamic environment. Experiments are conducted to show the effectiveness of the proposed method in real environment.**

I. Introduction

The recent advancement of MEMS technology makes the inertial measurement unit (IMU) small in size, low cost and power efficient. Thus, inertial navigation system (INS) based on an IMU have been widely used to estimate the trajectory of vehicle such as smart automobiles, micro air vehicles, robots, etc. However, due to error accumulation problem, it is very hard to get reliable result by only using the IMU. In order to tackle this problem, some INSs rely on GPS signal to periodically correct the IMU. But in some GPS-denied environments (e.g., indoors, underground, in tunnel, in space, etc.), this method does not work. Besides, the quality of GPS signal is affected by surrounding environment.

The camera is another choice to correct the IMU. Cameras are low cost, small size and can provide rich information about surrounding environment. By tracking feature points between several images, the motion of camera can be estimated [1]. The camera and the IMU are complementary sensors [2]. The IMU has lower uncertainty of measurement at fast motion and the camera can track feature points precisely at slow motion. Furthermore, the IMU can provide real scale which usually cannot be obtained by only using single camera. Based on above reasons, a monocular camera and an IMU are used for sensor fusion to estimate the ego-motion in this paper. Sensor fusion between the camera and the IMU is a popular research topic. It can provide many applications such as real scale estimation [3] [4], image de-blurring, IMU assisted feature tracking, ground plane detection, ego-motion estimation [5]-[9], etc. During the last two years, this research topic was trying to solve the initial condition problems [10].

The main purpose of this paper is the ego-motion estimation. Therefore, we propose an odometer architecture which combines a monocular camera and an IMU. The camera geometry constraints between three images are used as camera measurement information, which makes the proposed method without estimating the 3D position of feature point. Meanwhile, the camera pose corresponding to each of the three images are refined in filter to form a multi-state constraint Kalman filter (MSCKF). Consequently, the proposed method is a sliding window odometry which has a balance between computational cost and accuracy. In order to effectively reject feature points which are mismatch or fall on independently moving objects, the random sample consensus (RANSAC) algorithm based on three views geometry is used to choose inliers.

This paper is organized as follows. Related work is described in Section II. In Section III we explain our approach for visual assisted IMU odometer in detail. Experiments are described in Section IV. Finally, Section V concludes this paper.

II. Related Work

In general, there are two ways that the ego-motion can be estimated in map, which are visual odometry and visual simultaneous localization and mapping (SLAM) [11]. Visual odometry is a technique that estimates the ego-motion by using single or multiple cameras. The term visual odometry was first proposed by Nister et al. [12]. The reason for its name is similar with wheel odometry that both are estimating the motion trajectory by using the measurement information of sensors. Visual SLAM can be treated as corresponding research of visual odometry. In visual SLAM, Davison et al. first proposed one method of visual SLAM which only use single camera, and they named it monoSLAM [13]. In order to solve the depth estimation problem caused by parametrization of feature points in monoSLAM, Civera et al. proposed inverse depth parametrization to describe feature points [14]. After that, Civera et al. further proposed 1-point RANSAC method to reject unsuitable feature points [15]. The main difference between visual SLAM and visual odometry is that the goal of visual SLAM is keeping tracking the map of the environment, while visual odometry aims at estimating the motion trajectory incrementally. Considering our application which is the motion trajectory estimation, this paper focuses

Jwu-Sheng Hu is with the Institute of Electrical Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, ROC. and Mechanical and Systems Research Laboratories, Industrial Technology Research Institute (E-mail: jshu@cn.nctu.edu.tw).

Ming-Yuan Chen was with the Institute of Electrical Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, ROC. (E-mail: mychen.sid@gmail.com).

on visual odometry.

The most intuitive way to fuse IMU and monoSLAM is using the IMU motion model to replace the constant velocity motion model assumed in monoSLAM [5]. This method belongs to the tightly-coupled sensor fusion architecture. However, it is a SLAM-type method. The 3D position of feature point is estimated in the filter state vector that causes the following two problems: (1) the 3D position of feature point is assumed as Gaussian distribution and (2) as the number of feature points increase, the size of the filter state vector becomes enormous, resulting in more computational effort. In order to solve the above problems and focus on estimating the ego-motion, Mourikis et al. proposed a method that does not require estimating the 3D position of feature point in the filter state vector [6]. They use several past camera poses as the filter state vector. In measurement update step, the camera poses in the filter state vector are used to estimate the 3D position of feature point by a least square solution and then the 3D position of feature point is re-projected back to each camera pose which is used as the measurement model. Based on this, they proposed a MSCKF. The MSCKF is a sliding window odometry which has a balance between computational cost and accuracy.

In original MSCKF, it still requires a least square solution to estimate the 3D position of feature point. However, from the viewpoint of visual odometry, estimating the 3D point of feature point is not necessary, because it does not focus on reconstructing the map. In order to avoid estimating the 3D point of feature point, we use the camera geometry constraints (epipolar geometry and trifocal tensor) between three images as camera measurement information. In the literature, method that used trifocal tensor as the measurement model of Kalman filter can be traced back to the monocular visual odometry proposed by Ying-Kin et al. in 2006 [16]. Rather than only using epipolar geometry as measurement model [7], using trifocal tensor additionally can provide the consistent scale. After that, methods that use the camera geometry constraints as measurement information in different types of sensors were proposed (e.g., stereo camera [17], stereo camera with IMU [8], monocular camera with IMU [9], etc.). In this paper, we use the filter state design methodology of MSCKF and the camera geometry constraints between three images to propose a sliding window odometry which use a monocular camera and an IMU. In the following, we describe the algorithm in detail.

III. DESCRIPTION OF THE METHODOLOGY

The goal of the proposed method is to estimate the pose of the IMU in the global frame. Fig. 1 shows the geometric relationships between the IMU frame {I}, the camera frame {C} and the global frame {G}. The rotation matrix and position pair $({}^{I}\boldsymbol{R}_{C}, {}^{I}\boldsymbol{p}_{C})$ denotes the transformation of the camera frame with respect to the IMU frame. $({}^{G}\boldsymbol{R}_{I}, {}^{G}\boldsymbol{p}_{I})$ pair represents the transformation of the IMU frame in global frame. ${}^{G}\boldsymbol{p}_{Li}$ is the position of *i*-th feature point in the global frame.

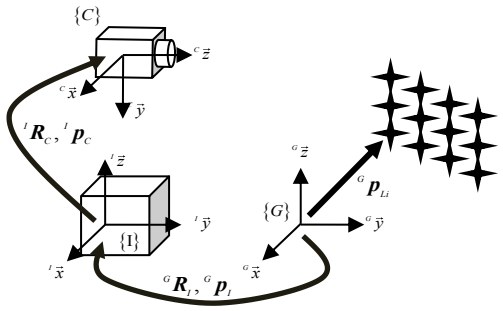

Fig. 1. IMU-Camera coordinates transition map

The IMU measures tri-axis acceleration and tri-axis angular velocity in real metric units. The measurements are given by the following equations:

$$\boldsymbol{a}_m = R^T\left({}^{G}\bar{\boldsymbol{q}}_I\right)\left({}^{G}\boldsymbol{a} + {}^{G}\boldsymbol{g}\right) + \boldsymbol{b}_a + \boldsymbol{n}_a \quad , \quad \boldsymbol{\omega}_m = {}^{I}\boldsymbol{\omega} + \boldsymbol{b}_g + \boldsymbol{n}_g \tag{1}$$

where $R\left({}^{G}\bar{\boldsymbol{q}}_I\right)$ is the direction cosine matrix (DCM) corresponding to quaternion ${}^{G}\bar{\boldsymbol{q}}_I$, ${}^{G}\boldsymbol{a}$ denotes the linear acceleration of the IMU with respect to the global frame, ${}^{G}\boldsymbol{g}$ is the gravitational acceleration in the global frame, ${}^{I}\boldsymbol{\omega}$ is angular velocity of the IMU in the IMU frame, $\boldsymbol{n}_a$ and $\boldsymbol{n}_g$ are modeled as white Gaussian noise, $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ are the biases of the accelerometer and gyroscope, respectively.

The state used to describe the IMU usually comprises its position, orientation, velocity and the biases of the accelerometer and the gyroscope. Let $\boldsymbol{x}_{IMU}$ be the IMU state:

$$\boldsymbol{x}_{IMU} = \begin{bmatrix} {}^{G}\boldsymbol{p}_I^{\,T} & {}^{G}\bar{\boldsymbol{q}}_I^{\,T} & {}^{G}\boldsymbol{v}_I^{\,T} & \boldsymbol{b}_a^{\,T} & \boldsymbol{b}_g^{\,T} \end{bmatrix}^T \tag{2}$$

The vector $\boldsymbol{x}_{IMU}$ is also called the IMU true state. In this paper, the biases $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ are modeled as Gaussian random walk process driven by $\boldsymbol{n}_{ba}$ and $\boldsymbol{n}_{bg}$, respectively. The IMU true-state kinematics describing the time evolution of the IMU state is given by the following equations [18]:

$$\begin{aligned} {}^{G}\dot{\boldsymbol{p}}_I = {}^{G}\boldsymbol{v}_I \quad , \quad {}^{G}\dot{\bar{\boldsymbol{q}}}_I = \frac{1}{2}{}^{G}\bar{\boldsymbol{q}}_I \otimes \begin{bmatrix} 0 & {}^{I}\boldsymbol{\omega}^T \end{bmatrix}^T \quad , \quad {}^{G}\dot{\boldsymbol{v}}_I = {}^{G}\boldsymbol{a} \\ \dot{\boldsymbol{b}}_a = \boldsymbol{n}_{ba} \quad , \quad \dot{\boldsymbol{b}}_g = \boldsymbol{n}_{bg} \end{aligned} \tag{3}$$

where $\otimes$ denotes quaternion multiplication. In order to minimize the dimension of the filter state vector and achieve the purpose of linearization, divide the IMU true state into nominal and error state:

$$\begin{aligned} {}^{G}\boldsymbol{p}_I = {}^{G}\hat{\boldsymbol{p}}_I + {}^{G}\tilde{\boldsymbol{p}}_I, \; {}^{G}\bar{\boldsymbol{q}}_I = {}^{G}\hat{\bar{\boldsymbol{q}}}_I \otimes \begin{bmatrix} 1 & \frac{1}{2}{}^{G}\delta\boldsymbol{\theta}_I^{\,T} \end{bmatrix}^T, \; {}^{G}\boldsymbol{v}_I = {}^{G}\hat{\boldsymbol{v}}_I + {}^{G}\tilde{\boldsymbol{v}}_I \\ \boldsymbol{b}_a = \hat{\boldsymbol{b}}_a + \tilde{\boldsymbol{b}}_a \quad , \quad \boldsymbol{b}_g = \hat{\boldsymbol{b}}_g + \tilde{\boldsymbol{b}}_g \end{aligned} \tag{4}$$

where $\left({}^{G}\hat{\boldsymbol{p}}_I, {}^{G}\hat{\bar{\boldsymbol{q}}}_I, {}^{G}\hat{\boldsymbol{v}}_I, \hat{\boldsymbol{b}}_a, \hat{\boldsymbol{b}}_g\right)$ and $\left({}^{G}\tilde{\boldsymbol{p}}_I, {}^{G}\delta\boldsymbol{\theta}_I, {}^{G}\tilde{\boldsymbol{v}}_I, \tilde{\boldsymbol{b}}_a, \tilde{\boldsymbol{b}}_g\right)$ are the IMU nominal state and the IMU error state, respectively. Since the mean of the noise is assumed as zero, the IMU nominal-state kinematics can be obtained by taking expectation of the IMU true-state kinematics (3):

$$
{}^{G}\dot{\hat{\boldsymbol{p}}}_I = {}^{G}\hat{\boldsymbol{v}}_I \quad , \quad {}^{G}\dot{\hat{\boldsymbol{q}}}_I = \frac{1}{2}\,{}^{G}\hat{\boldsymbol{q}}_I \otimes \hat{\boldsymbol{\omega}} \quad , \quad {}^{G}\dot{\hat{\boldsymbol{v}}}_I = R\left({}^{G}\hat{\boldsymbol{q}}_I\right)\hat{\boldsymbol{a}} - {}^{G}\boldsymbol{g} \qquad (5)
$$

$$
\dot{\hat{\boldsymbol{b}}}_a = \boldsymbol{0}_{3\times 1} \quad , \quad \dot{\hat{\boldsymbol{b}}}_g = \boldsymbol{0}_{3\times 1}
$$

with $\hat{\boldsymbol{a}} = \boldsymbol{a}_m - \hat{\boldsymbol{b}}_a$ and $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\boldsymbol{b}}_g$. By using the IMU true-state kinematics (3) and the IMU nominal-state kinematics (5), the IMU error-state kinematics can be obtained:

$$
\begin{aligned}
&{}^{G}\dot{\tilde{\boldsymbol{p}}}_I = {}^{G}\tilde{\boldsymbol{v}}_I \quad , \quad {}^{G}\delta\dot{\boldsymbol{\theta}}_I = -\lfloor \hat{\boldsymbol{\omega}} \times \rfloor\, {}^{G}\delta\boldsymbol{\theta}_I - \tilde{\boldsymbol{b}}_g - \boldsymbol{n}_g \\
&{}^{G}\dot{\tilde{\boldsymbol{v}}}_I = -R\left({}^{G}\hat{\boldsymbol{q}}_I\right)\lfloor \hat{\boldsymbol{a}} \times \rfloor\, {}^{G}\delta\boldsymbol{\theta}_I - R\left({}^{G}\hat{\boldsymbol{q}}_I\right)\tilde{\boldsymbol{b}}_a - R\left({}^{G}\hat{\boldsymbol{q}}_I\right)\boldsymbol{n}_a \\
&\dot{\tilde{\boldsymbol{b}}}_a = \boldsymbol{n}_{ba} \quad , \quad \dot{\tilde{\boldsymbol{b}}}_g = \boldsymbol{n}_{bg}
\end{aligned} \qquad (6)
$$

A. Structure of the filter state vector

The filter state vector comprises the IMU state and a history of last two poses of the camera. The filter state is also divided into nominal and error state. The filter nominal state is given by:

$$
\hat{\boldsymbol{x}}_k = \left[\, \hat{\boldsymbol{x}}_{IMU_k}{}^{T} \quad {}^{G}\hat{\boldsymbol{p}}_{I_1}{}^{T} \quad {}^{G}\hat{\boldsymbol{q}}_{I_1}{}^{T} \quad {}^{G}\hat{\boldsymbol{p}}_{I_2}{}^{T} \quad {}^{G}\hat{\boldsymbol{q}}_{I_2}{}^{T} \,\right]^{T} \qquad (7)
$$

where pair $\left({}^{G}\hat{\boldsymbol{p}}_{I_1}, {}^{G}\hat{\boldsymbol{q}}_{I_1}\right)$ denotes the nominal-state pose of the IMU corresponding to the last but one pose of the camera, while pair $\left({}^{G}\hat{\boldsymbol{p}}_{I_2}, {}^{G}\hat{\boldsymbol{q}}_{I_2}\right)$ is corresponding to the last pose of the camera. The filter error state is given by:

$$
\tilde{\boldsymbol{x}}_k = \left[\, \tilde{\boldsymbol{x}}_{IMU_k}{}^{T} \quad {}^{G}\tilde{\boldsymbol{p}}_{I_1}{}^{T} \quad {}^{G}\delta\boldsymbol{\theta}_{I_1}{}^{T} \quad {}^{G}\tilde{\boldsymbol{p}}_{I_2}{}^{T} \quad {}^{G}\delta\boldsymbol{\theta}_{I_2}{}^{T} \,\right]^{T} \qquad (8)
$$

Since past poses in filter prediction step has no dynamic, assume its process model is zero:

$$
\begin{aligned}
&{}^{G}\dot{\tilde{\boldsymbol{p}}}_{I_1} = 0 \;,\; {}^{G}\dot{\tilde{\boldsymbol{q}}}_{I_1} = 0 \quad \text{and} \quad {}^{G}\dot{\tilde{\boldsymbol{p}}}_{I_1} = 0 \;,\; {}^{G}\delta\dot{\boldsymbol{\theta}}_{I_1} = 0 \\
&{}^{G}\dot{\tilde{\boldsymbol{p}}}_{I_2} = 0 \;,\; {}^{G}\dot{\tilde{\boldsymbol{q}}}_{I_2} = 0 \qquad\quad {}^{G}\dot{\tilde{\boldsymbol{p}}}_{I_2} = 0 \;,\; {}^{G}\delta\dot{\boldsymbol{\theta}}_{I_2} = 0
\end{aligned} \qquad (9)
$$

B. Filter propagation

In filter prediction step, the nominal state use nominal-state kinematic (5)(9) with 4-th order Runge Kutta to predict. The prediction of error state is given by $\dot{\tilde{\boldsymbol{x}}}_k = \boldsymbol{F}_c\tilde{\boldsymbol{x}}_k + \boldsymbol{G}_c\boldsymbol{n}_{IMU}$. Digitize $\boldsymbol{F}_c$ to obtain $\boldsymbol{F}_d$ by using Taylor series:

$$
\boldsymbol{F}_d = \exp\left(\boldsymbol{F}_c\Delta t\right) = \boldsymbol{I}_{27} + \boldsymbol{F}_c\Delta t + \tfrac{1}{2!}\boldsymbol{F}_c{}^{2}\Delta t^{2} + \cdots \qquad (10)
$$

Analysis of the $\boldsymbol{F}_d$ matrix shows that its some elements have repetitive and sparse structure [4]. Therefore, without any approximation, it can be written as:

$$
\boldsymbol{F}_d = \begin{bmatrix}
\boldsymbol{I}_3 & \boldsymbol{\Phi}_{12} & \boldsymbol{\Phi}_{13} & \boldsymbol{\Phi}_{14} & \boldsymbol{\Phi}_{15} & \boldsymbol{0}_{3\times 12} \\
\boldsymbol{0}_{3\times 3} & \boldsymbol{\Phi}_{22} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{\Phi}_{25} & \boldsymbol{0}_{3\times 12} \\
\boldsymbol{0}_{3\times 3} & \boldsymbol{\Phi}_{32} & \boldsymbol{I}_3 & \boldsymbol{\Phi}_{34} & \boldsymbol{\Phi}_{35} & \boldsymbol{0}_{3\times 12} \\
\boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 12} \\
\boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 12} \\
\boldsymbol{0}_{12\times 3} & \boldsymbol{0}_{12\times 3} & \boldsymbol{0}_{12\times 3} & \boldsymbol{0}_{12\times 3} & \boldsymbol{0}_{12\times 3} & \boldsymbol{I}_{12}
\end{bmatrix} \qquad (11)
$$

Let $\boldsymbol{Q}_c$ be the noise covariance matrix:

$$
\boldsymbol{Q}_c = \boldsymbol{n}_{IMU}\boldsymbol{n}_{IMU}{}^{T} = diag\left(\sigma_g{}^{2}\cdot\boldsymbol{I}_3,\ \sigma_a{}^{2}\cdot\boldsymbol{I}_3,\ \sigma_{ba}{}^{2}\cdot\boldsymbol{I}_3,\ \sigma_{bg}{}^{2}\cdot\boldsymbol{I}_3\right) \qquad (12)
$$

where $\sigma_g{}^{2}$, $\sigma_a{}^{2}$, $\sigma_{ba}{}^{2}$ and $\sigma_{bg}{}^{2}$ is the variance of noises $\boldsymbol{n}_g$, $\boldsymbol{n}_a$, $\boldsymbol{n}_{ba}$ and $\boldsymbol{n}_{bg}$, respectively. Digitize $\boldsymbol{Q}_c$ to obtain $\boldsymbol{Q}_d$:

$$
\boldsymbol{Q}_d = \int_{\Delta t} \boldsymbol{F}_d(\tau)\boldsymbol{G}_c\boldsymbol{Q}_c\boldsymbol{G}_c{}^{T}\boldsymbol{F}_d(\tau)^{T}\, d\tau \qquad (13)
$$

With $\boldsymbol{F}_d$ and $\boldsymbol{Q}_d$, the prediction equation for error state covariance matrix $\boldsymbol{P}_{k|k}$ is given by $\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_d\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_d{}^{T} + \boldsymbol{Q}_d$.

C. Measurement update

The measurement model employed for updating the filter state estimate is given by the epipolar geometry and the trifocal tensor. As shown in Fig. 2, the epipolar geometry describes the geometry relationship between two images of the same static scene [1].

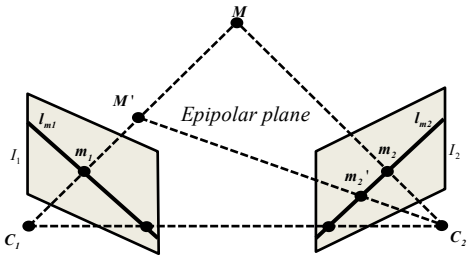

Fig. 2. the epipolar geometry between two views

The algebraic representation of epipolar geometry can be derived by the fundamental matrix $\boldsymbol{F}$. Let $\boldsymbol{t}$ and $\boldsymbol{R}$ be the position and rotation matrix of the frame C_2 with respect to the frame C_1. Then, the fundamental matrix is defined as $\boldsymbol{F} = \boldsymbol{K}^{-T}\boldsymbol{R}^{T}\lfloor \boldsymbol{t} \times \rfloor \boldsymbol{K}^{-1}$, where $\boldsymbol{K}$ is the camera intrinsic parameter matrix which can be obtained after camera calibration. Since the image point $\boldsymbol{m}_2'$ lies on the epipolar line $\boldsymbol{l}_{m2} = \boldsymbol{F}\boldsymbol{m}_1$, the epipolar geometry constraint is given by:

$$
\boldsymbol{m}_2{}^{T}\boldsymbol{F}\boldsymbol{m}_1 = 0 \qquad (14)
$$

The trifocal tensor encapsulates the geometry relationships between the three different viewpoints and is independent of the scene structure [1]. Fig. 3 shows the point-line-point correspondence between three views which can be used to transfer the image point by the trifocal tensor.

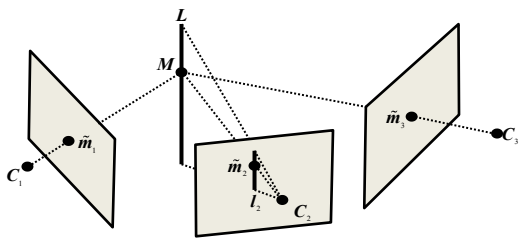

Fig. 3 the point-line-point correspondence between three views

Let $\boldsymbol{P}_1 = \left[\boldsymbol{I}|\boldsymbol{0}\right]$, $\boldsymbol{P}_2 = \left[\boldsymbol{A}|\boldsymbol{a}_4\right]$ and $\boldsymbol{P}_3 = \left[\boldsymbol{B}|\boldsymbol{b}_4\right]$ be the projection matrices of the camera at three different viewpoints $\boldsymbol{C}_1$,

$\boldsymbol{C}_2$ and $\boldsymbol{C}_3$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the 3×3 matrices, $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ denote the i-th column of the matrices $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$, respectively. According to the projection matrices and a line lies on 3D space, the trifocal tensor can be derived:

$$\boldsymbol{T}_i = \boldsymbol{b}_4\boldsymbol{a}_i^T - \boldsymbol{a}_4\boldsymbol{b}_i^T \tag{15}$$

Next, we use the trifocal tensor to transfer the point $\tilde{\boldsymbol{m}}_1$ in first frame and a line passing through the point $\tilde{\boldsymbol{m}}_2$ in second frame into the third frame which is called the point-line-point correspondence:

$$\tilde{\boldsymbol{m}}_3 = \left(\sum_i \tilde{m}_{1i}\boldsymbol{T}_i^T\right)\boldsymbol{l}_2 \tag{16}$$

The line $\boldsymbol{l}_2$ was recommended to choose as the line perpendicular to the epipolar line in [1].

The measurement model which comprises the epipolar geometry and the trifocal tensor is given by the following equations. Since we focus on consecutive camera pose, two epipolar constraints in three images are used. By assuming the correspondence of i-th feature point in three images is $\{\boldsymbol{m}_1,\boldsymbol{m}_2,\boldsymbol{m}_3\}$, the measurement value $\boldsymbol{z}_i$ is given by:

$$\boldsymbol{z}_i = h\left(\boldsymbol{x}_k,\{\boldsymbol{m}_1,\boldsymbol{m}_2,\boldsymbol{m}_3\}_i\right) = \begin{bmatrix} \tilde{\boldsymbol{m}}_2{}^T\boldsymbol{R}_{1,2}{}^T\lfloor\boldsymbol{t}_{12}\times\rfloor\tilde{\boldsymbol{m}}_1 \\ \tilde{\boldsymbol{m}}_3{}^T\boldsymbol{R}_{2,3}{}^T\lfloor\boldsymbol{t}_{23}\times\rfloor\tilde{\boldsymbol{m}}_2 \\ \boldsymbol{K}\left(\sum_i \tilde{\boldsymbol{m}}_{1i}\boldsymbol{T}_i^T\right)\boldsymbol{l}_2 \end{bmatrix} \tag{17}$$

with

$$\begin{aligned} &\boldsymbol{x}_k = g\left(\hat{\boldsymbol{x}}_k,\tilde{\boldsymbol{x}}_k\right)\ ,\ \tilde{\boldsymbol{m}}_1 = \boldsymbol{K}^{-1}\boldsymbol{m}_1\ ,\ \tilde{\boldsymbol{m}}_2 = \boldsymbol{K}^{-1}\boldsymbol{m}_2\ ,\ \tilde{\boldsymbol{m}}_3 = \boldsymbol{K}^{-1}\boldsymbol{m}_3 \\ &\boldsymbol{R}_{j,j+1} = \left({}^G\boldsymbol{R}_{C_j}\right)^T{}^G\boldsymbol{R}_{C_{j+1}}\ ,\ \boldsymbol{t}_{j,j+1} = \left({}^G\boldsymbol{R}_{C_j}\right)^T\left({}^G\boldsymbol{p}_{C_{j+1}} - {}^G\boldsymbol{p}_{C_j}\right)\ ,\ j=1,2 \\ &{}^G\boldsymbol{R}_{C_j} = R\left({}^G\bar{\boldsymbol{q}}_{I_j}\right){}^I\boldsymbol{R}_C\quad ,\ {}^G\boldsymbol{p}_{C_j} = {}^G\boldsymbol{p}_{I_j} + R\left({}^G\bar{\boldsymbol{q}}_{I_j}\right){}^I\boldsymbol{p}_C\quad ,\ j=1,2 \\ &{}^G\boldsymbol{R}_{C_3} = R\left({}^G\bar{\boldsymbol{q}}_I\right){}^I\boldsymbol{R}_C\quad ,\ {}^G\boldsymbol{p}_{C_3} = {}^G\boldsymbol{p}_I + R\left({}^G\bar{\boldsymbol{q}}_I\right){}^I\boldsymbol{p}_C \\ &\boldsymbol{l}_2 = \left(l_{e2},-l_{e1},-\tilde{m}_{2u}l_{e2}+\tilde{m}_{2v}l_{e1}\right)^T\ ,\ \boldsymbol{R}_{12}{}^T\lfloor\boldsymbol{t}_{12}\times\rfloor\tilde{\boldsymbol{m}}_1 = \left(l_{e1},l_{e2},l_{e3}\right)^T \\ &\tilde{\boldsymbol{m}}_2 = \left(\tilde{m}_{2u},\tilde{m}_{2v},1\right)^T \end{aligned} \tag{18}$$

where $\boldsymbol{x}_k$ is true state which is obtained by the nominal and error state according to (4). Since the measurement model is nonlinear, we use a Sigma-Point approach to update the filter state estimate. After measurement update, use error state $\tilde{\boldsymbol{x}}_{k|k}$ to correct nominal state and then obtain $\hat{\boldsymbol{x}}_{k|k}$. In order to keep only three poses in the filter state vector, replace old state by current state and revise error covariance:

$$\boldsymbol{T}_n = \begin{bmatrix} \boldsymbol{I}_{7\times7} & \boldsymbol{0}_{7\times9} & \boldsymbol{0}_{7\times7} & \boldsymbol{0}_{7\times7} \\ \boldsymbol{0}_{9\times7} & \boldsymbol{I}_{9\times9} & \boldsymbol{0}_{9\times7} & \boldsymbol{0}_{9\times7} \\ \boldsymbol{0}_{7\times7} & \boldsymbol{0}_{7\times9} & \boldsymbol{0}_{7\times7} & \boldsymbol{I}_{7\times7} \\ \boldsymbol{I}_{7\times7} & \boldsymbol{0}_{7\times9} & \boldsymbol{0}_{7\times7} & \boldsymbol{0}_{7\times7} \end{bmatrix}, \boldsymbol{T}_e = \begin{bmatrix} \boldsymbol{I}_{6\times6} & \boldsymbol{0}_{6\times9} & \boldsymbol{0}_{6\times6} & \boldsymbol{0}_{6\times6} \\ \boldsymbol{0}_{9\times6} & \boldsymbol{I}_{9\times9} & \boldsymbol{0}_{9\times6} & \boldsymbol{0}_{9\times6} \\ \boldsymbol{0}_{6\times6} & \boldsymbol{0}_{6\times9} & \boldsymbol{0}_{6\times6} & \boldsymbol{I}_{6\times6} \\ \boldsymbol{I}_{6\times6} & \boldsymbol{0}_{6\times9} & \boldsymbol{0}_{6\times6} & \boldsymbol{0}_{6\times6} \end{bmatrix}$$

$$\hat{\boldsymbol{x}}_k = \boldsymbol{T}_n\hat{\boldsymbol{x}}_k\ ,\ \tilde{\boldsymbol{x}}_k = \boldsymbol{T}_e\tilde{\boldsymbol{x}}_k\ ,\ \boldsymbol{P}_{k|k} = \boldsymbol{T}_e\boldsymbol{P}_{k|k}\boldsymbol{T}_e^T \tag{19}$$

D. RANSAC

In order to reject feature points which are mismatch or located on independently moving objects, we use RANSAC algorithm to select inliers. In Kalman filter, the procedure of RANSAC algorithm in this paper is similar to 1-point RANSAC EKF [15]. However, since the proposed method does not estimate the 3D position of feature point, it cannot use Euclidean distance to decide inlier by re-project the 3D position of feature point. Therefore, the proposed method uses the trifocal tensor to decide inlier:

$$\begin{aligned} &\{\boldsymbol{m}_1,\boldsymbol{m}_2,\boldsymbol{m}_3\}^{Inliers} \\ &= \left\{\{\boldsymbol{m}_1,\boldsymbol{m}_2,\boldsymbol{m}_3\}\ \middle|\ \left\|\boldsymbol{m}_3 - \boldsymbol{K}\left(\sum_i \tilde{\boldsymbol{m}}_{1i}\boldsymbol{T}_i^T\right)\boldsymbol{l}_2\right\| < threshold\right\} \end{aligned} \tag{20}$$

Since the trifocal tensor is derived under static assumption, the criterion (20) can detect the feature points located on independently moving objects which makes the overall algorithm capable of operating in dynamic environment.

E. Overall algorithm

The proposed visual assisted IMU odometer is summarized in **Algorithm 1**. It has the following characteristics: (1) tightly-coupled sensor fusion approach, (2) structure-less visual inertial odometry and (3) perform optimization over a sliding window of filter states.

Algorithm 1: Visual Assisted IMU Odometer

Initialize $\hat{\boldsymbol{x}}_{0|0}$, $\boldsymbol{P}_{0|0}$ and $\tilde{\boldsymbol{x}}_{0|0} = \boldsymbol{0}_{27\times1}$
for $k = 1,\cdots$ **do**
{ Time update }
  Compute $\boldsymbol{F}_d$ and $\boldsymbol{Q}_d$ by (11) and (13)
  %%% Propagate error state and error covariance %%%
  $\tilde{\boldsymbol{x}}_{k|k-1} = \boldsymbol{0}_{27\times1}$, $\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_d\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_d^T + \boldsymbol{Q}_d$
  Use 4-th order Runge Kutta method to predict $\hat{\boldsymbol{x}}_{k|k-1}$
{ Measurement update }
  if New image **then**
    Match feature points in last three images to get $\{m_1,m_2,m_3\}_i$
    Use RANSAC to find inliers
    %%% Generate sigma points and predict measurement %%%
    $\tilde{\boldsymbol{X}}^l_{k|k-1} = \boldsymbol{0}_{27\times1} \pm \left(\sqrt{(L+\lambda)\boldsymbol{P}_{k|k-1}}\right)_l$
    $\boldsymbol{Z}^l_i = h\left(g\left(\hat{\boldsymbol{x}}_{k|k-1},\tilde{\boldsymbol{X}}^l_{k|k-1}\right),\{\boldsymbol{m}_1,\boldsymbol{m}_2,\boldsymbol{m}_3\}_i\right)$, $\hat{\boldsymbol{z}}_i = \sum_{l=0}^{2L} W_s^l\boldsymbol{Z}^l_i$
    %%% update error state and error covariance%%%
    $\boldsymbol{P}_{z_iz_i} = \sum_{l=0}^{2L}\left(\boldsymbol{Z}^l_i - \hat{\boldsymbol{z}}_i\right)\left(\boldsymbol{Z}^l_i - \hat{\boldsymbol{z}}_i\right)^T + \boldsymbol{R}$
    $\boldsymbol{P}_{xz_i} = \sum_{l=0}^{2L} W_c^l\left(\tilde{\boldsymbol{X}}^l_{k|k-1} - \boldsymbol{0}_{27\times1}\right)\left(\boldsymbol{Z}^l_i - \hat{\boldsymbol{z}}_i\right)^T$
    $\boldsymbol{K}_k = \boldsymbol{P}_{xz_i}\boldsymbol{P}_{z_iz_i}{}^{-1}$
    $\tilde{\boldsymbol{x}}_{k|k} = \tilde{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\right)$, $\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{P}_{z_iz_i}\boldsymbol{K}_k{}^T$
    Use $\tilde{\boldsymbol{x}}_{k|k}$ to correct nominal state estimate and then obtain $\hat{\boldsymbol{x}}_{k|k}$
    %%% Replace old state and revise error covariance %%%
    $\hat{\boldsymbol{x}}_k = \boldsymbol{T}_n\hat{\boldsymbol{x}}_k$, $\tilde{\boldsymbol{x}}_k = \boldsymbol{T}_e\tilde{\boldsymbol{x}}_k$, $\boldsymbol{P}_{k|k} = \boldsymbol{T}_e\boldsymbol{P}_{k|k}\boldsymbol{T}_e^T$
  end if
end for

IV. Experimental Results

The proposed method is evaluated by using a publicly available real-world dataset [20]. The Matlab code of the proposed method can be downloaded from the internet [24]. In KITTI dataset, the sensor used for data recording consist of two grayscale and two color video cameras (Point Grey Flea2, 10 Hz, 1392×512 pixel resolution, $90^{o}\times35^{o}$ opening angle), a laser scanner and a GPS/IMU INS (OXTS RT 3003, 100 Hz). The proposed method only uses the measurements of a single grayscale camera and the IMU (acceleration and angular velocity) to estimate the ego-motion.

Geiger et al. [20] provided two versions of data which are raw and synchronized. After the manual synchronization, the IMU sampling rate is 10Hz. Since the synchronization between the camera and the IMU is important, we use the synchronized data to verify the proposed method. In the experiments, the extraction and matching of feature points are performed using the SIFT algorithm [21]. In order to maintain a certain amount of computational cost, we use the "bucketing" concept [17] to choose a subset of feature points. The initial velocity of the IMU and the initial direction of gravity were obtained from GPS/IMU INS. We use Euclidean distance and rotation angle to define position and orientation error and use a table to show overall RMSE and end point error. The proposed method is compared with the following methods: (1) GPS/IMU INS and use it as ground truth, (2) pure IMU navigation which is obtained by integrating acceleration and angular velocity and (3) monocular and stereo visual odometry proposed by Geiger et al. [22]. In the following, the experiments are conducted in three cases.

A. Case 1

The trajectory is about 540 m, takes 78 sec. The average speed is about 25 km/h. Fig. 4 shows the motion trajectory estimation results. It can be found that the result of the pure IMU navigation is not reliable with the IMU error accumulation. As shown in TABLE 1, the proposed method outperforms the other methods in the overall RMSE and end point error. It is worth noting that the angle from the pure IMU navigation is more close to the ground truth than the position. The main reason is that the angle is obtained by using single integration, while the position is obtained by using double integral. Furthermore, the orientation error would propagate to the position error in the gravity compensation step. The result of the monocular visual odometry is better than the pure IMU navigation. In general, the real scale cannot be obtained by only using single camera. In this monocular visual odometry, the real scale is derived by assuming that the camera is moving at a known and fixed height on the ground.

B. Case 2

The trajectory is about 2160 m, takes 94.5 sec. The average speed is about 82 km/h. The data in case 2 was acquired on the highway. Thus, the average speed in case 2 is much faster than case 1. The motion trajectory estimation results are shown in Fig. 5 and TABLE 2. It can be found that the result of the pure IMU navigation in case 2 is more reliable than in case 1. The main reason is that the IMU has lower uncertainty of measurement at fast motion which makes the better integration result.

C. Case 3

The trajectory is about 3577 m, takes 440 sec. The average speed is about 29 km/h. The path in this case is longer than in case 1 or case 2. Fig. 6 and TABLE 3 show the motion trajectory estimation results. After the three cases, we conclude that the motion trajectory estimated by the sensor fusion of two different sensors is more reliable.

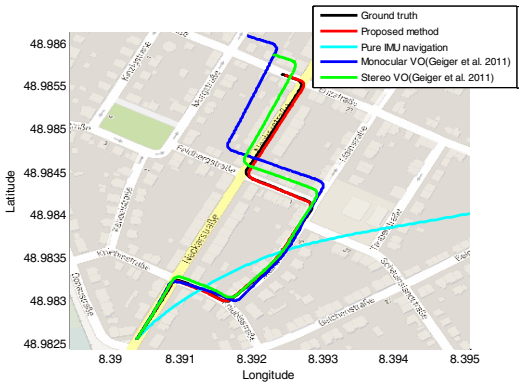

Fig. 4. the motion trajectory estimation results in KITTI dataset case1

TABLE 1. the overall RMSE and the end point error results in KITTI case 1

Algorithm	Overall position RMSE (m)	Overall orientation RMSE (deg)	End point position error (m)	End point orientation error (deg)
Proposed method	4.0018	1.1628	6.4478	1.0586
Pure IMU navigation	2748	11.1340	6009	9.4521
Monocular VO	33.9685	7.8149	67.5990	11.3223
Stereo VO	15.3520	4.0740	26.7258	6.3967

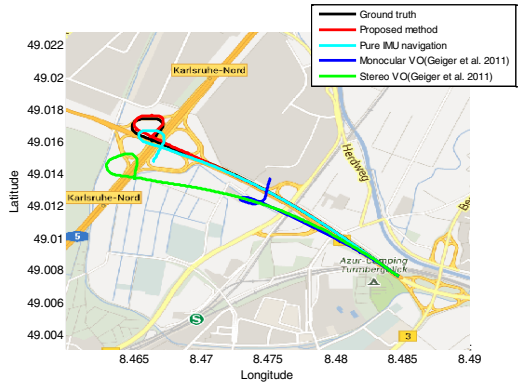

Fig. 5. the motion trajectory estimation results in KITTI case2

TABLE 2. the overall RMSE and the end point error results in KITTI case 2

Algorithm	Overall position RMSE (m)	Overall orientation RMSE (deg)	End point position error (m)	End point orientation error (deg)
Proposed method	34.2638	2.3190	28.3338	2.4629
Pure IMU navigation	64.8753	4.0528	147.6343	10.2659
Monocular VO	596.3744	96.4882	704.6405	164.5494
Stereo VO	215.7575	19.0431	300.1239	27.5330

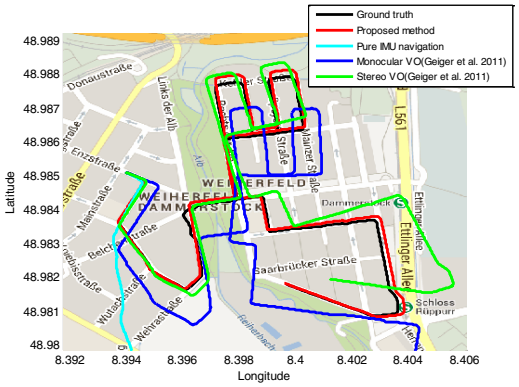

Fig. 6. the motion trajectory estimation results in KITTI case3

TABLE 3. the overall RMSE and the end point error results in KITTI case 3

Algorithm	Overall position RMSE (m)	Overall orientation RMSE (deg)	End point position error (m)	End point orientation error (deg)
Proposed	16.9207	0.8480	13.4773	0.8194
Pure IMU navigation	6742	4.7723	14731	10.0392
Monocular VO	211.2474	14.8340	304.5535	23.7477
Stereo VO	73.4203	10.6717	118.9049	17.7305

V. Conclusions

This paper presents an odometer architecture which combines a monocular camera and an IMU. The trifocal tensor geometry relationship between three images is used as camera measurement information, which makes the proposed method without estimating the 3D position of feature point. Meanwhile, the camera pose corresponding to each of the three images are refined in filter to form a MSCKF. The proposed method has the following characteristics: (1) tightly-coupled sensor fusion approach, (2) structure-less visual inertial odometry and (3) perform optimization over a sliding window of filter states. This paper further proposes a RANSAC algorithm which is based on three views geometry to select inliers. The experiments are conducted to show the effectiveness of the proposed method by using a publicly available real-world dataset. The results show the error of the IMU can be effectively constrained by the proposed method and the estimated ego-motion is close to the actual path.

Acknowledgment

This work was supported in part by the National Science Council, Taiwan, under grant # NSC 101-2221-E-009-002.

References

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*, 2nd ed. Cambridge University Press, 2008.

[2] Corke, J. Lobo, and J. Dias, "An Introduction to Inertial and Visual Sensing," *Intl. Journal of Robotics Research*, vol. 26, no. 6, pp. 519-535, Jun. 2007.

[3] H. Jwu-Sheng, T. Chin-Yuan, C. Ming-Yuan and S. Kuan-Chun, "IMU-Assisted Monocular Visual Odometry Including the Human Walking Model for Wearable Applications," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 6-10, 2013.

[4] S. Weiss and R. Siegwart, "Real-Time Metric State Estimation for Modular Vision-Inertial Systems," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13, 2011, pp. 4531-4537.

[5] P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardos, "Inertial Aiding of Inverse Depth SLAM using a Monocular Camera," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Roma, Italy, Apr. 10-14, 2007, pp. 2797-2802.

[6] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Roma, Italy, Apr. 10-14, 2007, pp. 3565-3572.

[7] J. O. Nilsson, D. Zachariah, M. Jansson, and P. Handel, "Realtime implementation of visual-aided inertial navigation using epipolar constraints," in *Proc. of the IEEE/ION Position Location and Navigation Symposium (PLANS)*, Myrtle Beach, SC, USA, Apr. 23-26, 2012, pp. 711-718.

[8] E. Asadi and C. L. Bottasso, "Tightly-coupled vision-aided inertial navigation via trifocal constraints," in *Proc. of the IEEE Intl. Conf. on Robotics and Biomimetics (ROBIO)*, Shenzhen, China, Dec. 12-14, 2012, pp. 85-90.

[9] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, "Real-Time Vision-Aided Localization and Navigation Based on Three-View Geometry," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2239-2259, Jul. 2012.

[10] A. Martinelli, "Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination," *IEEE Trans. on Robotics*, vol. 28, no. 1, pp. 44-60, Feb. 2012.

[11] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80-92, Dec. 2011

[12] D. Nister, O. Naroditsky, and J. Bergen, "Visual Odometry," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Washington, DC, USA, Jun. 27-Jul. 2, 2004, pp. 652-659.

[13] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, Jun. 2007.

[14] J. Civera, A. J. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Trans. on Robotics*, vol. 24, no.5, pp. 932-945, Oct. 2008.

[15] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-Point RANSAC for EKF Filtering. Application to Real-Time Structure from Motion and Visual Odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609-631, Sep. 2010.

[16] Y. Ying-Kin, W. Kin Hong, M. M. Y. Chang, and O. Siu Hang, "Recursive Camera-Motion Estimation With the Trifocal Tensor," *IEEE Trans. on Systems Man and Cybernetics Part B Cybernetics*, vol. 36, no. 5, pp. 1081-1090, Oct. 2006.

[17] B. Kitt, A. Geiger, and H. Lategahn, " Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme," in Proc. of the IEEE Intelligent Vehicles Symposium (IV), La Jolla, CA, USA, Jun. 21-24, 2010, pp. 486-492.

[18] J. Sola. (2012, Nov. 6). *Quaternion kinematics for the error-state KF* [Online]. Available: http://www.joansola.eu/JoanSola/objectes/notes/kinematics.pdf

[19] S. Julier and J. Uhlmann "A new extension of the Kalman filter to nonlinear systems", *Proc. SPIE 3068,* Orlando, FL, USA, April 21, 1997, pp.182 -193.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. of the IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, Jun. 16-21, 2012, pp. 3354-3361.

[21] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.

[22] A. Geiger, J. Ziegler, and C. Stiller, " StereoScan: Dense 3d Reconstruction in Real-time," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 5-9, 2011, pp. 963-968.

[23] J. Y. Bouguet. (2010, Jul. 9). *Camera Calibration Toolbox for Matlab* [Online].Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[24] http://www.mathworks.com/matlabcentral/fileexchange/43218-visualinertial-odometry