Advanced Robotics, 2015
Vol. 29, No. 20, 1289–1301, http://dx.doi.org/10.1080/01691864.2015.1057616

Taylor & Francis
Taylor & Francis Group

SURVEY PAPER

A review of visual inertial odometry from filtering and optimisation perspectives

Jianjun Gui*, Dongbing Gu, Sen Wang and Huosheng Hu

School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

(*Received 15 April 2015; accepted 25 May 2015*)

Visual inertial odometry (VIO) is a technique to estimate the change of a mobile platform in position and orientation overtime using the measurements from on-board cameras and IMU sensor. Recently, VIO attracts significant attentions from large number of researchers and is gaining the popularity in various potential applications due to the miniaturisation in size and low cost in price of two sensing modularities. However, it is very challenging in both of technical development and engineering implementation when accuracy, real-time performance, robustness and operation scale are taken into consideration. This survey is to report the state of the art VIO techniques from the perspectives of filtering and optimisation-based approaches, which are two dominated approaches adopted in the research area. To do so, various representations of 3D rigid motion body are illustrated. Then filtering-based approaches are reviewed, and followed by optimisation-based approaches. The links between these two approaches will be clarified via a framework of the Bayesian Maximum A Posterior. Other features, such as observability and self calibration, will be discussed.

Keywords: visual inertial odometry; SLAM; Kalman filtering; state estimation

1. Introduction

Localisation and mapping are two fundamental problems in the research area of robot navigation and control. This is evidenced by the fact that simultaneous localisation and mapping (SLAM) techniques and structure from motion (SFM) techniques [1] have been one of major topics in robotic and computer vision research communities, respectively, for many years. With the increased interest in applying these techniques to small-sized platforms, such as small-sized unmanned aerial vehicles (UAVs) or handhold mobile devices, the research focus of localisation and mapping is shifted towards the use of cameras and inertial measurement unit (IMU) sensor. These sensors are made available nowadays with high accuracy, miniaturised size, and low cost because of the fast-developing manufacturing of chips and micro-electromechanical systems (MEMS) devices. And they are complimentary with one another in a way which would be able to compensate for the errors made by each of them via the redundant information they provided. Furthermore, the evidence from biological studies shows that the navigation of human beings and some of animals is partly depending on various forms of the combination between motion sensing modalities and vision.[2–4]

Localisation and mapping problem is a state estimation problem in robotic community. Stochastic estimation algorithms for dynamic systems given noisy measurements, such as extended Kalman filter (EKF) or particle filters (PF), are the main stream tools being used. Some proprioceptive sensors provide the measurements on change in pose overtime and are formulated as a data-driven dynamic model of mobile platforms. The examples include optical encoders in ground mobile robots and IMUs in flying robots. They are well known for the accumulated errors and biased measurements. Some of exteroceptive sensors, such as cameras or laser ranging finders, are able to provide angular or range measurements. Based on triangulation or trilateration methods, they are able to estimate the position or orientation of mobile platforms. The estimation reliability heavily depends on environment conditions and the results are sensitive to the noise of measurements. However, they can be formulated as the measurement models in state estimation problems. Then, EKF or PF is able to propagate the estimated distribution from previous step to current step by fusing the distribution in dynamic model and the distribution in measurement models. In most cases, EKF is used instead of PF due to the large number of states to be estimated. Then, the means and covariances of Gaussian distributions are estimated.

Motion estimation from vision is an optimisation problem in computer vision community.[5] The change in camera pose is iteratively computed via image alignment.[6,7] Normally, features are extracted from one image and reprojected to another image captured from a different view. Then, the error cost between features and

*Corresponding author. Email: jgui@essex.ac.uk

reprojected features is minimised to find the pose change of camera and the structures of environment. The optimisation-based algorithms are mainly gradient based, such as Newton's method or Gauss–Newton's method. In most cases, the sparse structure of matrix is exploited to increase the computational efficiency. Only the values of estimated states are provided while the information on estimation distributions is not available.

The link between filtering and optimisation-based approaches can be built up within the framework of Bayesian inference. The optimisation-based approaches are viewed as a maximum likelihood (ML) formula where the state for which the total probability of measurements is highest is iteratively found. The filtering-based approaches are viewed as a Maximum A Posterior (MAP) formula, where the prior distribution of platform pose is constructed from the measurements of proprioceptive sensors and the likelihood distribution is built up with the measurements of exteroceptive sensors. For non-linear dynamic model and/or non-linear measurement model, iterated EKF is equivalent to the optimisation-based approaches where iterative updates on each single step in EKF are conducted. The optimisation-based approaches could be reformulated as a MAP problem from a ML problem by adding a regularisation term or prior term from the measurement of proprioceptive sensors or other sources. These are on-line or 'causal' algorithms where the current estimation depends on current and previous measurements. The off-line or 'non-casual' algorithms are batch processing procedures where the current estimation depends on full data-set or it can be said they depend on not only current and previous measurements but also future measurements. In optimisation-based approaches, the batch processing is melt down to solving a group of linear algebraic equations. In filtering-based approaches, the Kalman smoother is able to find the posterior Gaussian distribution via a forward pass and a backward pass.

There is a long list of state variables and parameters in a VIO problem, which needs to be estimated by given measurements from IMU and camera. Some of them are not observable or identifiable, leading to an error growing performance. The analysis of observability and identifiability of states and parameters provides a clear understanding of the estimation results, and potentially provides a guide to sufficiently exciting the platform via specialised motion patterns.

The popular SLAM techniques are able to provide the estimated results on pose of platform and structure of environment. A process called loop closure detection is required to bound the accumulated errors caused by estimation algorithms.[8–10] However, the loop closure is not explicitly pursued in VIO techniques. Therefore, the estimated errors in VIO would be accumulated and could not be bounded. Due to no loop closure detection, long time and large-scale localisation become possible within small-sized device. Then, the research question is shifted towards how to exploit the techniques that could slow down the error increasing.

This paper will present a survey of recent developments and advances of VIO techniques. It is organised from both filtering and optimisation perspectives, illustrating their fundamental models, algorithms and recent results. The survey will also highlight the links between two approaches and contribute an in-depth view of VIO techniques. The discussion on state observability and parameter identifiability will be provided. Our vision on this research area is summarised.

In the following, filtering-based approaches are introduced in Section 2, followed by optimisation-based approaches in Section 3. The links between them are summarised in Section 4. The analysis of observability and identifiability will be addressed in Section 5. The conclusions and future work are briefly provided in Section 6.

2. Filtering-based approaches

An EKF framework generally consists of a prediction step and an updating step. For a filtering-based VIO approach, inertial sensors are able to provide acceleration and rotational velocity measurements in three axes, which serve as the data-driven dynamic model or prior distribution for a 3D rigid body motion and make the motion prediction in prediction step. Cameras are able to provide the angular and ranging measurements between features and the mobile platform, which serve as the measurement model or likelihood distribution and update the prediction results in updating step.

We assume a mobile platform, which is only equipped with a camera and an IMU, moves in an unknown environment. The spatial relationship between the camera and IMU is fixed and can be expressed as known position and attitude. The aim of an EKF based VIO algorithm is to provide the information of position and orientation of the platform using inertial measurements and visual observations of unknown environment. Next, we will present a full description of the EKF framework based on the work in [11], which includes the state presentation, IMU data-driven dynamics and the visual observations. This will pave a way for further analysis and discussion in the following sections.

2.1. *IMU data driven dynamic model*

An IMU state vector of 3D rigid body at any time instant can be defined by a 16×1 vector,

$$\boldsymbol{x}_{\mathtt{I}} = \left[{}^{\mathtt{I}}_{\mathtt{W}}\bar{\boldsymbol{q}}^{\mathtt{T}} \;\; {}^{\mathtt{W}}\boldsymbol{p}_{\mathtt{I}}^{\mathtt{T}} \;\; {}^{\mathtt{W}}\boldsymbol{v}_{\mathtt{I}}^{\mathtt{T}} \;\; \boldsymbol{b}_g^{\mathtt{T}} \;\; \boldsymbol{b}_a^{\mathtt{T}} \right]^{\mathtt{T}}$$

where ${}^{\mathtt{I}}_{\mathtt{W}}\bar{\boldsymbol{q}}$ is the unit quaternion describing the rotation from world frame $\{\mathtt{W}\}$ to IMU frame $\{\mathtt{I}\}$, ${}^{\mathtt{W}}\boldsymbol{p}_{\mathtt{I}}$ and ${}^{\mathtt{W}}\boldsymbol{v}_{\mathtt{I}}$ are the position and velocity with respect to $\{\mathtt{W}\}$, $\boldsymbol{b}_g$ and $\boldsymbol{b}_a$ are 3×1 vectors that describe the biases affecting the gyroscope and accelerometer measurements, respectively. The spatial relations between frames are shown in Figure 1.

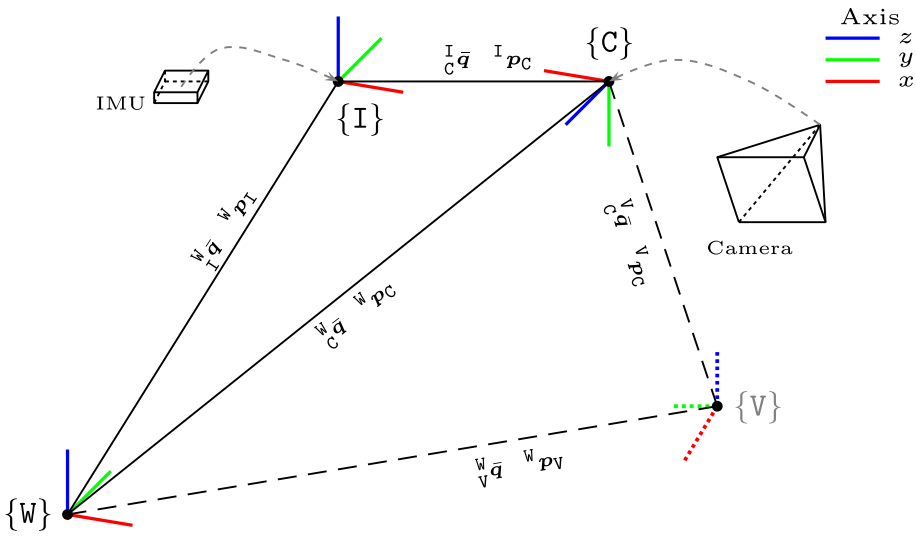

Figure 1. Coordinate frames. Each frame can be transformed from other frame by a rotation $\bar{\boldsymbol{q}}$ and a translation $\boldsymbol{p}$.

Assuming that the inertial measurements contain noises with zero-mean Gaussian models, denoted as $\boldsymbol{n}_g$ and $\boldsymbol{n}_a$, the real angular velocity $\boldsymbol{\omega}$ and the real acceleration $\boldsymbol{a}$ are related with gyroscope and accelerometer measurements in the following form:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \boldsymbol{b}_g + \boldsymbol{n}_g \quad \boldsymbol{a}_m = \boldsymbol{a} + \boldsymbol{b}_a + \boldsymbol{n}_a$$

The data-driven dynamic model is a combination of 3D rigid body dynamics and the above IMU measurements, which can be represented by the following form:

$$\begin{aligned} {}^{\mathtt{I}}_{\mathtt{W}}\dot{\bar{\boldsymbol{q}}}(\mathtt{t}) &= \frac{1}{2}\Omega(\boldsymbol{\omega}){}^{\mathtt{I}}_{\mathtt{W}}\bar{\boldsymbol{q}} \quad {}^{\mathtt{W}}\dot{\boldsymbol{v}}_{\mathtt{I}} = \mathrm{C}^{\mathrm{T}}_{{}^{\mathtt{I}}_{\mathtt{W}}\bar{\boldsymbol{q}}}\boldsymbol{a} - \boldsymbol{g} \\ {}^{\mathtt{W}}\dot{\boldsymbol{p}}_{\mathtt{I}} &= {}^{\mathtt{W}}\boldsymbol{v}_{\mathtt{I}} \quad \dot{\boldsymbol{b}}_g = \boldsymbol{n}_{wg} \quad \dot{\boldsymbol{b}}_a = \boldsymbol{n}_{wa} \end{aligned}$$

where $\mathrm{C}_{{}^{\mathtt{I}}_{\mathtt{W}}\bar{\boldsymbol{q}}}$ denotes a rotational matrix described by ${}^{\mathtt{I}}_{\mathtt{W}}\bar{\boldsymbol{q}}$ and $\boldsymbol{g}$ as the gravity vector in world frame $\{\mathtt{W}\}$. $\boldsymbol{\omega} = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^{\mathrm{T}}$ is the angular velocity expressed in IMU frame $\{\mathtt{I}\}$, while $\Omega(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathrm{T}} & 0 \end{bmatrix}$ is the quaternion kinematic matrix with $\lfloor\boldsymbol{\omega}\times\rfloor$ representing the skew-symmetric matrix. The IMU biases are modelled as random walk process, driven by the Gaussian noise, $\boldsymbol{n}_{wg}$ and $\boldsymbol{n}_{wa}$.

Let unit quaternion be $\bar{\boldsymbol{q}} := \left(q_0, \boldsymbol{q}^{\mathrm{T}}\right)^{\mathrm{T}}$ and its corresponding rotational matrix be $\mathrm{C}_{\bar{\boldsymbol{q}}}$. Two orientation representations can be linked via the equation below:

$$\mathrm{C}_{\bar{\boldsymbol{q}}} = \left(2q_0^2 - 1\right)\boldsymbol{I}_3 - 2q_0\lfloor\boldsymbol{q}\times\rfloor + 2\boldsymbol{q}\boldsymbol{q}^{\mathrm{T}}$$

Apart from the current IMU state $\boldsymbol{x}_{\mathtt{I}}$ mentioned above, a widely used state vector for filtering-based approaches includes a spatial scale λ (slow drift) and current camera pose $\left[{}^{\mathtt{W}}_{\mathtt{C}}\bar{\boldsymbol{q}}^{\mathrm{T}} \;\; {}^{\mathtt{W}}\boldsymbol{p}^{\mathrm{T}}_{\mathtt{C}}\right]^{\mathrm{T}}$. Here, we present a system state only containing a camera pose and it can be expressed as a 24-element vector. However, in various methods, the whole system state may include more than one past camera pose and keeps a moving window to limit them during state updating.

$$\boldsymbol{x} = \left[{}^{\mathtt{I}}_{\mathtt{W}}\bar{\boldsymbol{q}}^{\mathrm{T}} \;\; {}^{\mathtt{W}}\boldsymbol{p}^{\mathrm{T}}_{\mathtt{I}} \;\; {}^{\mathtt{W}}\boldsymbol{v}^{\mathrm{T}}_{\mathtt{I}} \;\; \boldsymbol{b}^{\mathrm{T}}_g \;\; \boldsymbol{b}^{\mathrm{T}}_a \;\; \lambda \;\; {}^{\mathtt{W}}_{\mathtt{C}}\bar{\boldsymbol{q}}^{\mathrm{T}} \;\; {}^{\mathtt{W}}\boldsymbol{p}^{\mathrm{T}}_{\mathtt{C}}\right]^{\mathrm{T}}$$

with other three differential equations,

$$\dot{\lambda} = 0 \;\; {}^{\mathtt{W}}_{\mathtt{C}}\dot{\bar{\boldsymbol{q}}} = 0 \;\; {}^{\mathtt{W}}\dot{\boldsymbol{p}}_{\mathtt{C}} = 0$$

Applying the expectation operator in above equations, we obtain the prediction results using the IMU data-driven dynamic model:

$$\begin{aligned} {}^{\mathtt{I}}_{\mathtt{W}}\dot{\hat{\bar{\boldsymbol{q}}}} &= \tfrac{1}{2}\Omega\left(\boldsymbol{\omega}_m - \hat{\boldsymbol{b}}_g\right){}^{\mathtt{I}}_{\mathtt{W}}\hat{\bar{\boldsymbol{q}}} \quad {}^{\mathtt{W}}\dot{\hat{\boldsymbol{v}}}_{\mathtt{I}} = \mathrm{C}^{\mathrm{T}}_{{}^{\mathtt{I}}_{\mathtt{W}}\hat{\bar{\boldsymbol{q}}}}\left(\boldsymbol{a}_m - \hat{\boldsymbol{b}}_a\right) - \boldsymbol{g} \\ {}^{\mathtt{W}}\dot{\hat{\boldsymbol{p}}}_{\mathtt{I}} &= {}^{\mathtt{W}}\hat{\boldsymbol{v}}_{\mathtt{I}} \quad \dot{\hat{\boldsymbol{b}}}_g = \boldsymbol{0} \quad \dot{\hat{\boldsymbol{b}}}_a = \boldsymbol{0} \quad \dot{\hat{\lambda}} = 0 \quad {}^{\mathtt{W}}_{\mathtt{C}}\dot{\hat{\bar{\boldsymbol{q}}}} = \boldsymbol{0} \quad {}^{\mathtt{W}}\dot{\hat{\boldsymbol{p}}}_{\mathtt{C}} = \boldsymbol{0} \end{aligned} \tag{1}$$

This can be abstracted as a non-linear system function with camera measurement $\boldsymbol{z}$ and a process noise term $\boldsymbol{n}_p \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$,

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{z}_k) + \boldsymbol{n}_{p,k} \tag{2}$$

2.2. *Error state representation and updating*

For the position, velocity and bias state variables, the arithmetic difference can be applied (i.e. the error in the estimate $\hat{\boldsymbol{x}}$ of a quantity $\boldsymbol{x}$ is defined as $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$), but the error quaternion should be defined under the assumption as local minimal situation.[12] If $\hat{\bar{\boldsymbol{q}}}$ is the estimated value of quaternion $\bar{\boldsymbol{q}}$, then the orientation error is described by the error quaternion $\delta\bar{\boldsymbol{q}}$, which is defined by the relation $\bar{\boldsymbol{q}} = \delta\bar{\boldsymbol{q}} \otimes \hat{\bar{\boldsymbol{q}}} \Rightarrow \delta\bar{\boldsymbol{q}} = \bar{\boldsymbol{q}} \otimes \hat{\bar{\boldsymbol{q}}}^{-1}$. In this expression, the symbol $\otimes$? denotes quaternion multiplication. Intuitively, the quaternion $\delta\bar{\boldsymbol{q}}$ describes a small rotation that causes the true and estimated attitude to coincide. Since attitude

corresponds to three Degree of Freedom (DoF), $\delta\theta$ is used to describe the attitude errors, which is a minimal representation. The error quaternion $\delta\bar{\boldsymbol{q}}$ can be written as

$$\delta\bar{\boldsymbol{q}} = \begin{bmatrix} \frac{1}{2}\delta\theta \\ \sqrt{1 - \frac{1}{4}\delta\theta^{\mathrm{T}}\delta\theta} \end{bmatrix} \approx \begin{bmatrix} \frac{1}{2}\delta\theta \\ 1 \end{bmatrix}$$

Thus, the error state vector containing 22 elements is

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \delta\theta_{\mathtt{W}}^{\mathtt{I}\mathrm{T}} & {}^{\mathtt{W}}\tilde{\boldsymbol{p}}_{\mathtt{I}}^{\mathrm{T}} & {}^{\mathtt{W}}\tilde{\boldsymbol{v}}_{\mathtt{I}}^{\mathrm{T}} & \tilde{\boldsymbol{b}}_g^{\mathrm{T}} & \tilde{\boldsymbol{b}}_a^{\mathrm{T}} & \tilde{\lambda} & \delta\theta_{\mathtt{C}}^{\mathtt{W}\mathrm{T}} & {}^{\mathtt{W}}\tilde{\boldsymbol{p}}_{\mathtt{C}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

The differential equations for the continuous time error state are

$$\begin{aligned} \delta\dot{\theta}_{\mathtt{W}}^{\mathtt{I}} &= -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor\delta\theta_{\mathtt{W}}^{\mathtt{I}} - \tilde{\boldsymbol{b}}_g - \boldsymbol{n}_g \\ {}^{\mathtt{W}}\dot{\tilde{\boldsymbol{v}}}_{\mathtt{I}} &= -\mathrm{C}_{\mathtt{I}_{\hat{\mathtt{W}}}\hat{\boldsymbol{q}}}^{\mathrm{T}}\left(\lfloor\hat{\boldsymbol{a}}\times\rfloor + \tilde{\boldsymbol{b}}_a + \boldsymbol{n}_a\right) \\ {}^{\mathtt{W}}\dot{\tilde{\boldsymbol{p}}}_{\mathtt{I}} &= {}^{\mathtt{W}}\tilde{\boldsymbol{v}}_{\mathtt{I}} \quad \dot{\tilde{\boldsymbol{b}}}_g = \boldsymbol{n}_{wg} \quad \dot{\tilde{\boldsymbol{b}}}_a = \boldsymbol{n}_{wa} \\ \dot{\tilde{\lambda}} &= 0 \quad {}_{\mathtt{C}}^{\mathtt{W}}\dot{\tilde{\boldsymbol{q}}} = \boldsymbol{0} \quad {}^{\mathtt{W}}\dot{\tilde{\boldsymbol{p}}}_{\mathtt{C}} = \boldsymbol{0} \end{aligned}$$

with $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\boldsymbol{b}}_g$ and $\hat{\boldsymbol{a}} = \boldsymbol{a}_m - \hat{\boldsymbol{b}}_a$.

By stacking the differential equations for error state, the linearised continuous time error state equation can be formed,

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{F}_c\tilde{\boldsymbol{x}} + \boldsymbol{G}_c\boldsymbol{n}$$

with the noise vector $\boldsymbol{n} = \left[\boldsymbol{n}_a^{\mathrm{T}}, \boldsymbol{n}_{wa}^{\mathrm{T}}, \boldsymbol{n}_g^{\mathrm{T}}, \boldsymbol{n}_{wg}^{\mathrm{T}}\right]^{\mathrm{T}}$. And the covariance matrix of $\boldsymbol{n}$ depends on the noise characteristics of IMU, $\boldsymbol{Q}_{\mathtt{C}} = \mathrm{diag}\left(\boldsymbol{\sigma}_{\boldsymbol{n}_a}^2, \boldsymbol{\sigma}_{\boldsymbol{n}_{wa}}^2, \boldsymbol{\sigma}_{\boldsymbol{n}_g}^2, \boldsymbol{\sigma}_{\boldsymbol{n}_{wg}}^2\right)$.

In practical, the inertial measurements for state propagation are obtained from IMU in discrete form, thus we assume the signals from gyroscopes and accelerometers are sampled with time interval Δt, and the state estimate is propagated using numerical integration like Runge–Kutta methods. Moreover, the covariance matrix of EKF filter is defined as

$$\boldsymbol{P}_{k|k} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{II}_{k|k}} & \boldsymbol{P}_{\boldsymbol{IC}_{k|k}} \\ \boldsymbol{P}_{\boldsymbol{IC}_{k|k}} & \boldsymbol{P}_{\boldsymbol{CC}_{k|k}} \end{bmatrix}$$

where $\boldsymbol{P}_{\boldsymbol{II}_{k|k}}$ is the covariance matrix of the IMU state, $\boldsymbol{P}_{\boldsymbol{CC}_{k|k}}$ is the $6N \times 6N$ covariance matrix of the camera pose estimates and $\boldsymbol{P}_{\boldsymbol{IC}_{k|k}}$ is the correlation between errors in IMU state and camera pose estimates. With this notation, the covariance matrix can be propagated by

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_d\boldsymbol{P}_{k|k}\boldsymbol{F}_d^{\mathrm{T}} + \boldsymbol{Q}_d \tag{3}$$

where the state-transition matrix can be calculated by assuming $\boldsymbol{F}_c$ and $\boldsymbol{G}_c$ to be constant over the integration time interval between two consecutive steps,

$$\boldsymbol{F}_d = \exp(\boldsymbol{F}_c\Delta t) = \boldsymbol{I} + \boldsymbol{F}_c\Delta t + \frac{1}{2}\boldsymbol{F}_c^2\Delta t^2 + \cdots \tag{4}$$

and the discrete-time covariance matrix $\boldsymbol{Q}_d$ can also be derived through numerical integration,

$$\boldsymbol{Q}_d = \int_{\Delta t} \boldsymbol{F}_d(\tau)\,\boldsymbol{G}_c\,\boldsymbol{Q}_c\,\boldsymbol{G}_c^{\mathrm{T}}\boldsymbol{F}_d(\tau)^{\mathrm{T}}\,d\tau \tag{5}$$

Thus, the mean and covariance propagation process as of the EKF-based VIO framework is summarized as follows,

(a) when IMU data, $\boldsymbol{\omega}_m$ and $\boldsymbol{a}_m$, in a certain sample frequency, are available to the filter, the state vector is propagated using numerical integration on Equation (1).
(b) calculate $\boldsymbol{F}_d$ and $\boldsymbol{Q}_d$ according to (4) and (5) respectively.
(c) the propagated state covariance matrix is computed from (3).

2.3. *Visual measurement model and updating*

Due to the biases and noises in IMU data, the prediction results from prediction step become worse and worse with time increasing. The measurements from visual sensors would be able to provide key information to bound the increased errors. To do so in an EKF framework, key information extracted from images should be cast into measurement equations. There are various methods to build a measurement model. For example, the loosely coupled methods where image alignment is used to directly obtain the position and orientation changes fuse two estimated results together via an EKF framework.[13] The so-called tightly coupled methods advocate the use of key information extracted from images. The key information could be the features extracted from images via feature detectors,[14] direct light intensity with depth information (point cloud),[15,16] or semi-direct light intensity with depth information.[17] The key information is modelled as the measurement equation so that an analytic relationship between the key information and the state variables is provided. In general, a non-linear algebraic equation can be viewed as the measurement equation:

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{n}_{m,k} \tag{6}$$

where $\boldsymbol{n}_m$ models the Gaussian noise with zero mean and covariance $\boldsymbol{R}$ in visual measurement.

After linearisation, the measurement error is expressed in a linear form:

$$\tilde{\boldsymbol{z}} = \hat{\boldsymbol{z}} - \boldsymbol{z} = \boldsymbol{H}\tilde{\boldsymbol{x}} + \boldsymbol{n}_m \tag{7}$$

where $\boldsymbol{H}$ is the Jacobian matrix, and the noise term is Gaussian distributed and uncorrelated to the state error. $\hat{\boldsymbol{z}}$ and $\boldsymbol{z}$ represent the prediction and real measurement, respectively.

For using features as the visual information, the most straightforward way is to exploit the pinhole model to build up the measurement equation in camera frame $\{\mathtt{C}\}$. A point $\boldsymbol{u} = [u \;\; v]^{\mathrm{T}}$ in an image is extracted to represent the key information. First, the spatial relationship between a point $\boldsymbol{u}$ in the camera frame and a point ${}^{\mathtt{W}}\boldsymbol{p}_{\boldsymbol{u}}$ in the world frame $\{\mathtt{W}\}$ is established as following equation:

$$[x \;\; y \;\; z]^{\mathrm{T}} = {}^{\mathtt{C}}\boldsymbol{p}_{\boldsymbol{u}} = \mathrm{C}_{\mathtt{C}_{\hat{\mathtt{W}}}\tilde{\boldsymbol{q}}}\left({}^{\mathtt{W}}\boldsymbol{p}_{\boldsymbol{u}} - {}^{\mathtt{W}}\boldsymbol{p}_{\mathtt{C}}\right)\lambda$$

Strictly speaking, the visual world $\{\mathtt{V}\}$ is different to the real world $\{\mathtt{W}\}$, but they are linked by initialising fixed

translation ${}^{\mathbb{W}}\boldsymbol{p}_{\mathrm{V}}$ and rotation ${}^{\mathbb{W}}\bar{\boldsymbol{q}}_{\mathrm{V}}$ as shown in Figure 1. Here, for simplicity, we only use the world frame $\{\mathbb{W}\}$.

The position in the image frame is linked to the state variables ${}^{\mathbb{W}}\boldsymbol{p}_{\mathrm{C}}$ and ${}^{\mathbb{W}}\bar{\boldsymbol{q}}_{\mathrm{C}}$ in the world frame via the camera pinhole model, described in Equation (10). The measurement model thus can be expressed briefly as

$$z = \boldsymbol{u} = \frac{1}{z}[x \quad y]^{\mathrm{T}} + \boldsymbol{n}_m$$

And the estimation of this point in the world frame ${}^{\mathbb{W}}\hat{\boldsymbol{p}}_{\boldsymbol{u}}$ can be calculated by iterative minimisation methods beforehand.[18]

For other key visual information, such as pixel intensity or gradient, the measurement model should build up the link between a measurement and the state variables, then the non-linear measurement equation is linearised to obtain the Jacobian matrix.

At this point, the updating step is ready for updating the prediction results made in prediction step. The Kalman gain is calculated as

$$\boldsymbol{K} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{H}\boldsymbol{P}_{k+1|k}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1} \tag{8}$$

and final correction $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{K}\cdot\tilde{z}$. After the correction, we can get the updated state estimate $\boldsymbol{x}_k$. Lastly, the error state covariance is updated as

$$\boldsymbol{P}_{k+1|k+1} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\,\boldsymbol{P}_{k+1|k} \tag{9}$$

The full update process is summarised as follows:

(a) when visual data, normally raw image, in a certain sample frequency are available, some image processing procedures are adopted to extract the key information.
(b) calculate the residual as (7).
(c) compute the Kalman gain as (8).
(d) update the state by adding the correction.
(e) update the error state covariance as (9).

3. Optimisation-based approaches

The optimisation-based approaches mainly rely on the techniques of image processing for feature extraction and optimisation for image alignment, while inertial measurement is treated as prior, regularisation terms or totally ignored. In most cases, there are two stages in an optimisation-based approach: mapping and tracking.[15] In mapping stage, features, such as corners, edges or other landmarks in 3D space, are extracted from an image via various features detectors. Then a reprojected error is defined between two images for all the features detected. The error is used as a cost function to be optimised in order to find the coordinates of features or landmarks.[7] In tracking stage, the coordinates of features and landmarks in the map are used to define a reprojected error between two images, and an optimisation algorithm is applied again to find the changes in position and orientation of the mobile platform. The idea of separating the estimation problem into two stages is to obtain a fast tracking result while the mapping processing is time consuming.[14,15] Simultaneously optimising a cost defined using reprojected error between two images against coordinates of 3D features and pose changes of the mobile platform is possible [19] while using the concept of keyframes is able to marginalise old states to maintain a bounded optimisation window for real-time operation.

3.1. Feature alignment

Iterative non-linear optimisation is formulated to find the camera pose changes and/or feature coordinates by minimising a reprojection error of observed regions in images. Normally, a map consists of features identified in a number of keyframes in which significant features are found. The map is presented by a series of 3D coordinate vectors of features. When a new image is obtained, a decision should be made about whether or not it is a keyframe. If so, the coordinates of new features found in this new image is computed via an image alignment algorithm and added to the map together with current camera pose. Otherwise, the map keeps unchanged.

As shown in Figure 2, a 3D rigid body transformation $\boldsymbol{T} \in SE(3)$ denotes rotation and translation in 3D:

$$\boldsymbol{T} = \begin{pmatrix} \mathsf{R} & \mathsf{t} \\ 0 & 1 \end{pmatrix} \quad \text{with} \quad \mathsf{R} \in SO(3) \quad \text{and} \quad \mathsf{t} \in \mathbb{R}^3$$

The optimisation purpose in image alignment is to find the transformation $\boldsymbol{T}$ in each time step, i.e. $\boldsymbol{T}$ is regarded as the camera pose. A minimal representation for camera pose is better for optimisation purpose. The Lie algebra $se(3)$ corresponding to the tangent space of $SE(3)$ at the identity is used as the minimal representation. The algebra element is called twisted coordinates $\boldsymbol{\xi} = [\boldsymbol{\omega}^{\mathrm{T}} \quad \boldsymbol{v}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$. The map from Lie algebra $se(3)$ to Lie group $SE(3)$ is the exponential map $\boldsymbol{T}(\boldsymbol{\xi}) = \exp(\psi(\boldsymbol{\xi}))$ and its inverse map is the logarithm map $\psi(\boldsymbol{\xi}) = \log \boldsymbol{T}(\boldsymbol{\xi})$, where $\psi(\boldsymbol{\xi})$ is the wedge operator,

$$\psi(\boldsymbol{\xi}) = \begin{pmatrix} \lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{v} \\ \mathbf{0} & 1 \end{pmatrix}$$

A 3D point with homogeneous vector ${}^{\mathrm{C}}\boldsymbol{p}_{\boldsymbol{u}}$ in the camera frame maps to the image coordinate $\boldsymbol{u}$ via the pinhole camera projection model:

$$\boldsymbol{u} = \pi\left({}^{\mathrm{C}}\boldsymbol{p}_{\boldsymbol{u}}\right) = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix}\begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \end{pmatrix} \tag{10}$$

where u_0, v_0 and f_u, f_v are the principal point and focal length, respectively, representing camera intrinsic parameters which can be calibrated beforehand. Given a depth information $d_{\boldsymbol{u}}$ for a point $\boldsymbol{u}$, the 3D point in the camera frame can be recovered from an image coordinate:

$${}^{\mathrm{C}}\boldsymbol{p}_{\boldsymbol{u}} = \pi^{-1}(\boldsymbol{u}, d_{\boldsymbol{u}})$$

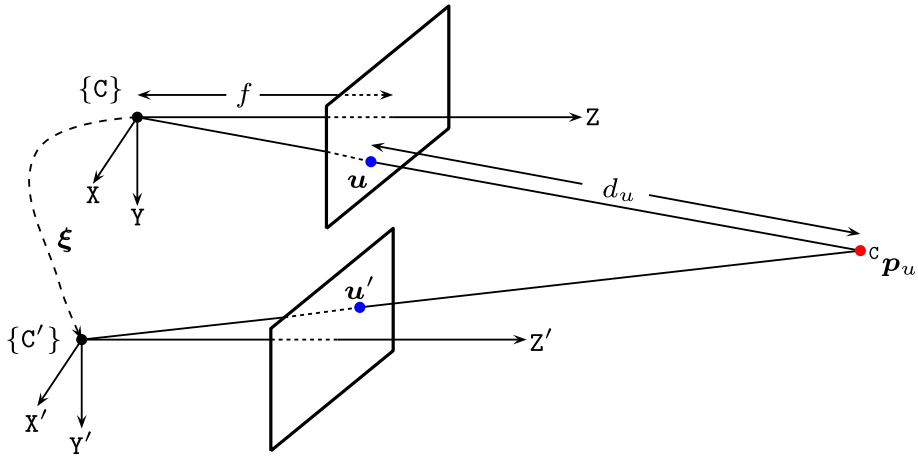

Figure 2. Projection. A 3D point ${}^{\mathrm{W}}\boldsymbol{p}_u$ is projected into two image frames linked by motion $\boldsymbol{\xi}$.

The mapping from a point ${}^{\mathrm{W}}\boldsymbol{p}_u$ in the world frame to a point in the camera frame is ${}^{\mathrm{C}}\boldsymbol{p}_u = \exp(\psi(\boldsymbol{\xi}))^{\mathrm{W}}\boldsymbol{p}_u$. For monocular camera, the visual measurement model becomes $z = [u \;\; v]^T = \boldsymbol{u}(\boldsymbol{\xi}, {}^{\mathrm{W}}\boldsymbol{p}_u)$. The depth information of a pixel is obtained by triangulating two consecutive keyframes using a set of independent filter.[14,20] For stereo cameras, $z = [u_l \;\; v_l \;\; u_r]^T$, which is also a function of $\boldsymbol{\xi}$ and ${}^{\mathrm{W}}\boldsymbol{p}_u$. The depth information is obtained via stereo vision techniques.

The reprojection error is defined as the difference between a measurement z and its estimate $\hat{z}(\hat{\boldsymbol{\xi}}, {}^{\mathrm{W}}\hat{\boldsymbol{p}}_u)$:

$$\tilde{z} = z - \hat{z}(\hat{\boldsymbol{\xi}}, {}^{\mathrm{W}}\hat{\boldsymbol{p}}_u)$$

The cost function $\eta\left(\hat{\boldsymbol{\xi}}, {}^{\mathrm{W}}\hat{\boldsymbol{p}}_u\right)$ is the sum of all squared errors $\tilde{z}$ with a weighting matrix $\boldsymbol{W}$:

$$\eta\left(\hat{\boldsymbol{\xi}}, {}^{\mathrm{W}}\hat{\boldsymbol{p}}_u\right) = \sum_{i=1}^{n}\sum_{j=1}^{m} \tilde{z}_{i,j}^{\mathrm{T}} w_{i,j} \tilde{z}_{i,j}$$

where j from 1 to m is the index of points within a frame, and i is the number of frames indexing a set with size n.

The optimisation problem is defined as

$$\boldsymbol{\xi}, {}^{\mathrm{W}}\boldsymbol{p}_u = \arg\min_{\boldsymbol{\xi}, {}^{\mathrm{W}}\boldsymbol{p}_u} \eta\left(\hat{\boldsymbol{\xi}}, {}^{\mathrm{W}}\hat{\boldsymbol{p}}_u\right)$$

The method used to solve the above problem is called Bundle Adjustment.[7]

In PTAM,[14] the optimisation process is separated into two parallel threads: mapping and tracking. Given the tracking results $\hat{\boldsymbol{\xi}}$, the optimisation problem in the mapping thread is:

$${}^{\mathrm{W}}\boldsymbol{p}_u = \arg\min_{{}^{\mathrm{W}}\boldsymbol{p}_u} \eta\left(\hat{\boldsymbol{\xi}}, {}^{\mathrm{W}}\boldsymbol{p}_u\right)$$

Given the mapping results ${}^{\mathrm{W}}\hat{\boldsymbol{p}}_u$, the optimisation problem in the tracking thread is:

$$\boldsymbol{\xi} = \arg\min_{\boldsymbol{\xi}} \eta\left(\boldsymbol{\xi}, {}^{\mathrm{W}}\hat{\boldsymbol{p}}_u\right)$$

In Levenberg–Marquardt (LM) algorithm, the key is to find an increment $\boldsymbol{\delta}$ in each iterative step, then update the optimised state variables. The solution to $\boldsymbol{\delta}$ is found by:

$$\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H} + \alpha\boldsymbol{I}\right)\boldsymbol{\delta} = -\boldsymbol{H}_e^{\mathrm{T}}\boldsymbol{W}\boldsymbol{z}$$

where $\boldsymbol{H}$ is the Jacobian of z and α is the LM damping parameter.

In most cases, only the features in sparse keyframes are maintained in the computer in order to limit the optimisation complexity as the same scale as filtering-based approaches.[19]

3.2. *Dense alignment*

The reprojection error is defined using the coordinates of features or landmarks within images in the above discussion. However, it requires a feature detection process, which ignores most parts of an image as the features are only extracted sparsely. The feature extraction process is often badly conditioned, noisy and not robust therefore relying on higher level robust estimation techniques. Since all these estimation steps are not on the level of raw image measurements (intensities), they systematically propagate feature extraction errors and accumulate drifts.

Appearance and optical flow-based techniques, on the other hand, are image-based and minimise an error directly based on raw image measurements,[21] i.e. the photometric(brightness or intensity) function is used, and therefore are called direct or dense methods. Dense methods aim at using the whole image for alignments. Non-linear optimisation techniques are used to find the transformation between two scenes. It is increasingly clear that it is possible to get more complete, accurate and robust results using dense methods for both mapping and tracking. The work in [22,23] minimises the photometric error via image alignment for visual odometry using Kinect RGB-D cameras. The work

in [24] includes both of photometric error and depth error to minimise.

With RGB-D camera available, such as Kinect sensors, the depth information of a pixel makes the alignment of multiple scans possible by minimising distance measures between all of the data in each image, rather than limited number of features or landmarks. Such dense scan is able to reconstruct surface scenes in the environment and track the pose. Given two successive dense depth measurements, surfaces or 3D point clouds of the same static scene observed from different viewpoints, one can find the change in pose of the camera by obtaining the rigid transformation that best maps one point cloud onto the other.

The Iterative Closest Point (ICP) is popular algorithm to match the scans through optimising the rigid transformation.[25] The ICP works in this way: given two corresponding point sets: $\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_n\}$ and $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_n\}$ and the translation and rotation between them are t and R, respectively, t and R are found by minimising the sum of squared errors:

$$\frac{1}{n}\sum_{i=1}^{n} \|\beta_i - \mathsf{R}\gamma_i - \mathsf{t}\|^2$$

If the correct correspondences are known, the rotation and translation can be calculated in a closed form. If correct correspondences are unknown as in most cases, it is generally impossible to find the optimal relative rotation and translation in one step. The iterating to find the alignment [25] is sought via finding the closest points' correspondence. The ICP converges if starting positions are close enough. One work using ICP for visual odometry is reported in [26].

The dense alignment can also be done by matching current image against a scene model. KinectFusion in [27] is the one which provides a real-time solution for dense alignment via separated mapping and tracking threads in GPU where a truncated signed distance function is employed for scene model. However, performing ICP on the full point cloud is computationally expensive, and does not provide a real-time solution on a general PC.

For a monocular camera, the depth information is not available. An inverse depth map is estimated in [15] by minimising a photometric error via non-convex optimisation algorithm in the mapping thread. In the tracking thread, the image alignment is used. Estimating the depth information in a monocular camera is also conducted via a Bayesian filter followed by an optimisation process to smooth the depth map in [28].

To further simplify the computational complexity and maintain the accuracy of dense methods, semi-dense or semi-direct methods have been proposed recently [29,30] for monocular camera. A semi-dense depth map covering all image regions with non-negligible gradients. The inverse depth map is estimated using the Bayesian filter while the tracking is obtained by directly minimising the dense photometric error. As stated in [30], semi-direct or semi-dense methods use hundreds of small patches to increase the robustness and allow for neglecting the patch normals.

3.3. Inertial measurement term

The measurement from an IMU sensor is the data source to the data-driven 3D rigid motion dynamics in filtering-based approaches, and is fused with the measurement from cameras via Kalman filter. This is tightly coupled as the cross variances between two parts are taken into consideration.[11] The loosely coupled fusion is to maintain a constant processing time by fusing the already estimated pose from visual sensor with the predicted pose from IMU via an EKF.[31] In optimisation-based approaches, the fusion between two parts can also be made tightly, i.e. no explicit pose estimates from camera is required. The predicted result from IMU driven dynamics in Equation (1) is viewed as a Gaussian distribution. The error between predicted results and true state is cast as the square error weighted by the covariance, then added to the cost function as a regularisation term.[19] In term of the Bayesian inference, the regularisation term is viewed as the prior while the image alignment term as the likelihood. The optimised result is a posteriori distribution, which is the one resulted from the ML estimation result of pure image alignment smoothed by the IMU prior.

4. Links between filtering and optimisation based approaches

Both of filtering-based and optimisation-based approaches can be formed under the Bayesian inference. When the succession of approximation linearisation is necessary, their link can be made explicitly via the iterated EKF. When the approximation linearisation is just a single step, the smoother-based approaches which include a forward pass and a backward pass are equivalent to the optimisation-based approaches which are solved via the Cholesky decomposition of information matrix of least square problems. To reduce the computational complexity, reducing the state variables to be estimated is implemented by maintaining only keyframes or a sliding window. In particular, the sliding window scheme or moving horizon estimation divides the cost into two parts, and one popular way to implement two parts is optimising the result in one part and marginalising out oldest states in another part with an EKF.

4.1. Iterated EKF update

The core of filtering based approaches is the Kalman filter and the core of optimisation-based approaches is the Gauss–Newton method. The link between them is the iterated EKF (IEKF).[32] An EKF has two steps: prediction and update. Let the result of prediction step is $\hat{\boldsymbol{x}}_k \sim \mathcal{N}(\boldsymbol{x}_k, \boldsymbol{P}_k)$ at current time k. The difference between EKF and IEKF is

that there is an iterative loop in the update step of IEKF while a single loop is executed in the update step of EKF. It is the iterative loop of IEKF which is able to drive the error caused by model linearisation as close as possible to the counterpart in optimisation-based approaches.

In the following, we will show the equivalence between the iterative loop in update step of IEKF and the Gauss–Newton method of optimisation approaches from maximising likelihood (ML).

At time k, the IEKF has $\boldsymbol{x}_k$, $\hat{\boldsymbol{x}}_{k|k-1}$ and $\boldsymbol{z}_k$ as the current state, the current state estimate and the measurement, respectively. The measurement model is the same as Equation (6) and $\hat{\boldsymbol{x}}_{k|k-1} \sim \mathcal{N}(\boldsymbol{x}_k, \boldsymbol{P}_{k|k-1})$.

Define an error vector as in quadratic cost function with a free variable $\boldsymbol{\mu}$.

$$\boldsymbol{e}(\boldsymbol{\mu}) = \boldsymbol{S}\begin{bmatrix} \boldsymbol{z}_k - h(\boldsymbol{\mu}) \\ \hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{\mu} \end{bmatrix}$$

where $\boldsymbol{S}^T\boldsymbol{S} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{k|k-1} \end{bmatrix}^{-1}$.

The maximum likelihood optimisation problem is:

$$\boldsymbol{\mu} = \arg\max_{\boldsymbol{\mu}} \exp\left(-\frac{1}{2}\boldsymbol{e}(\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{e}(\boldsymbol{\mu})\right)$$

or

$$\boldsymbol{\mu} = \arg\min_{\boldsymbol{\mu}} \left(\frac{1}{2}\boldsymbol{e}(\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{e}(\boldsymbol{\mu})\right)$$

Given the initial value $\boldsymbol{\mu}^{(0)} = \hat{\boldsymbol{x}}_{k|k-1}$, the Gauss–Newton method

$$\boldsymbol{\mu}^{(i+1)} = \boldsymbol{\mu}^{(i)} - \left(\left(\nabla \boldsymbol{e}(\boldsymbol{\mu}^{(i)})\right)^{\mathrm{T}} \nabla \boldsymbol{e}(\boldsymbol{\mu}^{(i)})\right)^{-1} \left(\nabla \boldsymbol{e}(\boldsymbol{\mu}^{(i)})\right)^{\mathrm{T}} \boldsymbol{e}(\boldsymbol{\mu}^{(i)})$$

And

$$\nabla \boldsymbol{e}(\boldsymbol{\mu}^{(i)}) = -\boldsymbol{S}\begin{bmatrix} \boldsymbol{H}^{(i)} \\ \boldsymbol{I} \end{bmatrix}$$

where $\boldsymbol{H}^{(i)} = \nabla h(\boldsymbol{\mu}^{(i)})$. Using the above gradient, the Gauss–Newton method becomes

$$\begin{aligned}\boldsymbol{\mu}^{(i+1)} &= \left(\boldsymbol{H}_{(i)}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H}^{(i)} + \boldsymbol{P}_{k|k-1}^{(i)}\right)^{-1} \\ &\quad \times \left(\boldsymbol{H}_{(i)}^{\mathrm{T}}\boldsymbol{R}^{-1}\left(\boldsymbol{z}_k - h(\boldsymbol{\mu}^{(i)}) + \boldsymbol{H}^{(i)}\boldsymbol{\mu}^{(i)}\right) + \boldsymbol{P}_{k|k-1}^{(i)}\hat{\boldsymbol{x}}_{k|k-1}\right) \\ &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k^{(i)}\left(\boldsymbol{z}_k - h(\boldsymbol{\mu}^{(i)}) - \boldsymbol{H}^{(i)} \times \left(\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{\mu}^{(i)}\right)\right) \end{aligned} \tag{11}$$

with the gain

$$\boldsymbol{K}_k^{(i)} = \boldsymbol{P}_{k|k-1}^{(i)}\boldsymbol{H}_{(i)}^{\mathrm{T}}\left(\boldsymbol{H}^{(i)}\boldsymbol{P}_{k|k-1}^{(i)}\boldsymbol{H}_{(i)}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1} \tag{12}$$

And the covariance is

$$\begin{aligned}\boldsymbol{P}_{k|k-1}^{(i+1)} &= \mathrm{E}\left[\left(\boldsymbol{\mu}^{(i+1)} - \boldsymbol{\mu}^{(i)}\right)^{\mathrm{T}}\left(\boldsymbol{\mu}^{(i+1)} - \boldsymbol{\mu}^{(i)}\right)\right] \\ &= \left(\boldsymbol{I} - \boldsymbol{K}_k^{(i)}\boldsymbol{H}^{(i)}\right)\boldsymbol{P}_{k|k-1}^{(i)} \end{aligned} \tag{13}$$

After the loop in i, it can be seen that the results from the update step are $\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{\mu}^{(i+1)}$ and $\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1}^{(i+1)}$.

In summary, the above iterative loop of Gauss–Newton method is viewed as the update step of IEKF, i.e.

(a) Initialization: $i = 0$, $\boldsymbol{\mu}^{(0)} = \hat{\boldsymbol{x}}_{k|k-1}$ and $\boldsymbol{P}_{k|k-1}^{(0)} = \boldsymbol{P}_{k|k-1}$;
(b) Loop calculation from Equations (11) to (13);
(c) Final updating: $\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{\mu}^{(i+1)}$ and $\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1}^{(i+1)}$.

When only one iterative loop is executed, the above is the update step of EKF. Their relationship can be viewed clearly in Figure 3.

4.2. *Smoothing-based approaches*

The filtering-based approaches are a sequence iterative process where the current estimated results depend on the current measurements and the past estimated results (the past measurements are condensed into the past estimated results), i.e. future measurements do not make any contributions to the current estimated results. If this is not the case, i.e. future measurements do make contributions to the current estimated results, the smoothing-based approaches should be applied. This is more like a batch algorithm where the estimation is made when all the measurements are available.[33]

The IMU data-driven dynamics (2) and the camera measurement Equation (6) are still adopted. The full probability is

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x}_0)\prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})\prod_{i=1}^{m} p(\boldsymbol{z}_i|\boldsymbol{x}_i)$$

The MAP problem is

$$\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{z}) = \arg\min_{\boldsymbol{x}}\left(\sum_{i=1}^{n}\|\boldsymbol{x}_i - f(\boldsymbol{x}_{i-1})\|_{\boldsymbol{P}_i} + \sum_{i=1}^{m}\|\boldsymbol{z}_i - h(\boldsymbol{x}_i)\|_{\boldsymbol{R}_i}\right)$$

Linearising $f(\cdot)$ and $h(\cdot)$, the above minimisation problem is rewritten as a group of linear equations in the general form:

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{\delta}^{\star} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$$

This can be solved by Cholesky decomposition of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$.[33] This is the perspective of optimisation-based approaches to the batch processing.

On the other hand, the batch processing can be interpreted from the perspective of filtering-based approaches, which is based on stochastic treatment.

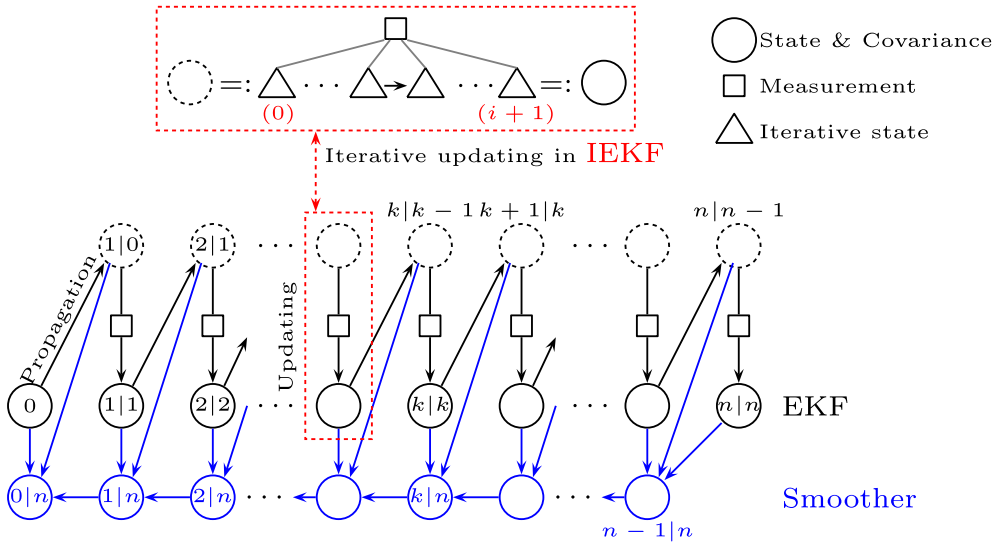

Figure 3. Links of methods. The process and relationship of EKF, IEKF and smoother.

The Kalman smoother experiences forward and backward passes, as shown in Figure 3. The forward pass computes $p(\boldsymbol{x}_k|z_{0:k})$ given the data available so far. The backward pass computes $p(\boldsymbol{x}_k|z_0, \cdots, z_n)$ given all the data.

Traditionally, the Kalman filter is expressed as

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+1|k} &= \mathrm{E}\left[\boldsymbol{x}_{k+1}|z_{0:k}\right] \\
\boldsymbol{P}_{k+1|k} &= \mathrm{E}\left[(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k})(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k})^{\mathrm{T}}|z_{0:k}\right] \\
\hat{\boldsymbol{x}}_{k|k} &= \mathrm{E}\left[\boldsymbol{x}_k|z_{0:k}\right] \\
\boldsymbol{P}_{k|k} &= \mathrm{E}\left[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})^{\mathrm{T}}|z_{0:k}\right]
\end{aligned}$$

After linearising, it becomes:

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{F}\hat{\boldsymbol{x}}_{k|k} \\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{F}\boldsymbol{P}_{k|k}\boldsymbol{F}^T + \boldsymbol{Q} \\
\boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1|k}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{k+1|k}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \\
\hat{\boldsymbol{x}}_{k+1|k+1} &= \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}(z_{k+1} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k+1|k}) \\
\boldsymbol{P}_{k|k} &= \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1}\boldsymbol{H}\boldsymbol{P}_{k+1|k}
\end{aligned}$$

The $p(\boldsymbol{x}_k|z_{0:k})$ only takes into account the past information relative to $\boldsymbol{x}_k$. If we incorporate it with the future observations, more refined state estimates can be found. Estimators that take into account both past and future are often called smoother, which can be written as following.

$$\begin{aligned}
\boldsymbol{L}_k &= \boldsymbol{P}_{k|k}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}_{k+1|k}^{-1} \\
\hat{\boldsymbol{x}}_{k|n} &= \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{L}_k\left(\hat{\boldsymbol{x}}_{k+1|n} - \hat{\boldsymbol{x}}_{k+1|k}\right) \\
\boldsymbol{P}_{k|n} &= \boldsymbol{P}_{k|k} + \boldsymbol{L}_k(\boldsymbol{P}_{k+1|n} - \boldsymbol{P}_{k+1|k})\boldsymbol{L}_k^{\mathrm{T}}
\end{aligned}$$

As an extension to Kalman smoother, it is possible to use an EM algorithm to learn the parameters of the system. The parameters include the noise covariances $\boldsymbol{Q}$, $\boldsymbol{R}$ and the $\boldsymbol{F}$, $\boldsymbol{H}$.

Although the above algorithm goes through two passes, there is no attempt to solve a succession of linear approximation to the non-linear problem. If there is no good linearisation point, the bad result would be expected. Also as the trajectory goes longer and longer, the scale of the linearised equations is unmanaged.

4.3. *Marginalisation to keyframes*

If all the state variables and features encountered during the course of operation are maintained, like the above scenario, the computational complexity becomes larger and larger with the increasing of exploring trajectory. However, not all the sensory data are useful. There are some of them which contain key information on tracking and mapping, while others could be redundant or contain trivial information which could be ignored. There are two ways to select useful information from the past historical data. One is a fixed window scheme in which only recent N data measurements are kept while all the data before N measurements are simply marginalised out. Another one is to select N data measurements from all the data-set according to some criteria, which is illustrated in Figure 4. The most popular criteria are the critical information contained in the data so that those keyframes are maintained during the entire processing.[19,34]

Marginalising out states is equivalent to applying the Schur compliment to the least squared problem. For example,

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^{\mathrm{T}} & \boldsymbol{A}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}$$

Changing to

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{0} & \boldsymbol{A}_{22} - \boldsymbol{A}_{12}^T\boldsymbol{A}_{11}\boldsymbol{A}_{12} \end{bmatrix}\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 - \boldsymbol{A}_{12}^T\boldsymbol{A}_{11}\boldsymbol{b}_1 \end{bmatrix}$$

After this forward substitution step, the smaller system is

$$\boldsymbol{A}_{22} - \boldsymbol{A}_{12}^T \boldsymbol{A}_{11} \boldsymbol{A}_{12} \boldsymbol{x}_2 = \boldsymbol{b}_2 - \boldsymbol{A}_{12}^T \boldsymbol{A}_{11} \boldsymbol{b}_1$$

Marginalising out the state $\boldsymbol{x}_1$ will induce dependencies between other states that are dependent on $\boldsymbol{x}_1$ like $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{12}$. And marginalising out the oldest pose from the full solution may cause fill-in in three places [35]:

- between any landmarks that are visible from that pose;
- between the states of the next oldest pose;
- between the next oldest pose and all landmarks seen by the removed pose.

The matrices in the problem solution would become dense and affect the computational efficiency.

4.4. *Moving horizon estimation*

Moving horizon estimation (MHE) approaches separates the MAP cost into two parts. Only recent N terms from step k to $\kappa = k - N + 1$ are optimised each step, while the oldest terms are summarised into an arrival cost, which is approximated by an EKF, as shown in Figure 4.

Let the time intervals be $[0, k-N]$ and $[\kappa, k]$. The MAP problem is

$$\arg\min_{\boldsymbol{x}} \left(\sum_{i=0}^{k-N} \|\boldsymbol{x}_{i+1} - f(\boldsymbol{x}_i)\|_{\boldsymbol{Q}}^2 + \sum_{i=0}^{k-N} \|z - h(\boldsymbol{x}_i)\|_{\boldsymbol{R}}^2 + \sum_{i=\kappa}^{k-1} \|\boldsymbol{x}_{i+1} - f(\boldsymbol{x}_i)\|_{\boldsymbol{Q}}^2 + \sum_{i=\kappa}^{k} \|z - h(\boldsymbol{x}_i)\|_{\boldsymbol{R}}^2 \right)$$

With the sum of first N terms is an arrival cost, the MHE problem is

$$\arg\min_{\boldsymbol{x}_\kappa, \boldsymbol{w}_\kappa^{k-1}} \left(\sum_{i=\kappa}^{k-1} \|\boldsymbol{n}_{p,i}\|_{\boldsymbol{Q}}^2 + \sum_{i=\tau}^{k} \|z - h(\boldsymbol{x}_i)\|_{\boldsymbol{R}}^2 + \phi_\kappa(\boldsymbol{x}_\kappa) \right)$$
$$\text{s.t.} \quad g(\boldsymbol{x}_i, \boldsymbol{z}_i, \boldsymbol{n}_{p,i}) \leq \boldsymbol{d}, \qquad i = \kappa, \cdots, k$$

where κ refers to the starting time of MHE window, ϕ_κ is the arrival cost at time $\tau \in [\kappa, k]$. The arrival cost can be approximated by an EKF:

$$\phi_\kappa(\boldsymbol{x}_\kappa) = \|\boldsymbol{x}_\kappa - \hat{\boldsymbol{x}}_\kappa\|_{\boldsymbol{P}_\kappa}^2$$

where the covariance $\boldsymbol{P}$ is propagated by

$$\boldsymbol{P}_{i+1} = \boldsymbol{Q} + \boldsymbol{F}(\boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{H}^{\mathrm{T}} (\boldsymbol{R} + \boldsymbol{H} \boldsymbol{P}_i \boldsymbol{H}^{\mathrm{T}})^{-1} \boldsymbol{H} \boldsymbol{P}_i) \boldsymbol{F}^{\mathrm{T}}$$

5. State observability and parameter identifiability

The problem which VIO is tackling with is to recover the platform trajectory in the global frame on-line given the measurements from inertial and visual sensors. However, this problem is solved not straightforward in terms of practical implementation. Apart from the complexity of filtering- or optimisation-based approaches, some parameters play a crucial role in the success of dynamic state estimation representing the platform trajectory. These parameters include:

- Camera intrinsic parameters: focal length, principal points and lens distortion.
- IMU parameters: acceleration and gyroscope biases.
- Spatial parameters: the transform between IMU and camera.
- Temporal parameter: the time delay between IMU and camera measurements

They are treated as time-invariant variables except the IMU biases and called parameters of the system. In contrast, the platform trajectory is represented by time-variant variables: states.

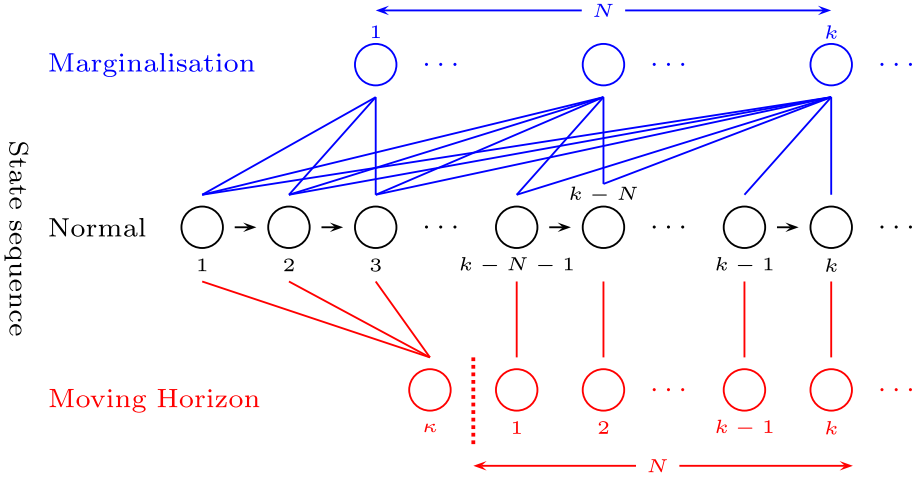

Figure 4. Keep N states. Marginalisation and moving horizon estimation.

Given only the measurements from IMU and camera, the question whether these states and parameters can be recovered can be answered from the analysis of observability and identifiability. Traditionally, states are dynamic variables and their estimation problem is analysed using observability while parameters are static variables and their calibration is analysed using identifiability. Modelling the parameters with random walk processes is able to analyse them using observability, like the IMU biases.

5.1. State observability

Observability is a fundamental property which reflects the possibility of estimating states on the basis of input/output data. Let $y\,(t, t_0, \boldsymbol{x}_0, z(t))$ denotes the output trajectory from an initial state $\boldsymbol{x}_0$, initial time t_0 and measurement reading $z(t)$ for the continuous time system. Two initial states $\boldsymbol{x}_0^1$ and $\boldsymbol{x}_0^2$ are defined to be indistinguishable if $y\left(t, t_0, \boldsymbol{x}_0^1, z(t)\right) = y\left(t, t_0, \boldsymbol{x}_0^2, z(t)\right)$ for $t_0 < t < t_N$, and all admissible measurement $z(t)$. If states are distinguishable, they can be estimated from the outputs and the known measurements. If states are indistinguishable, they are called unobservable and their corresponding variance will grow without bound and blow up.

Observability of a linear system is a global property that can be determined either from the rank of the observability matrix or from the rank of Gramian matrix. However, observability of a non-linear system is determined locally about a given state.[36] The local observability is stronger than the observability. The local observability distinguishes states from their neighbours. The local weak observability instantaneously distinguishes states from their neighbours without large excursions.[37]

The advantage of local weak observability is the availability of Lie derivative algebraic test. For a non-linear system composed by Equations (2) and (6). The Lie derivative of $h(\cdot)$ with respect to $f(\cdot)$ is

$$L_f h(\boldsymbol{x}) = \nabla h(\boldsymbol{x}) f(\boldsymbol{x})$$

The Lie derivatives can be defined recursively,

$$L_f^2 h(\boldsymbol{x}) = L_f(L_f h(\boldsymbol{x})) = \nabla L_f h(\boldsymbol{x}) f(\boldsymbol{x})$$
$$L_{f_i f_j}^2 h(\boldsymbol{x}) = L_{f_j}(L_{f_i} h(\boldsymbol{x})) = \nabla L_{f_i} h(\boldsymbol{x}) f_j(\boldsymbol{x})$$

The zero-order Lie derivative of any function is the function itself $L^0 h(\boldsymbol{x}) = h(\boldsymbol{x})$.

The observability matrix $\boldsymbol{O}$ is formed by stacking the Lie derivatives of $h(\boldsymbol{x})$ as its rows.

$$\boldsymbol{O} = \{L_{f_i \ldots f_j}^l h(\boldsymbol{x}) | i, j = 1, \ldots, l; l \in \mathbb{N}\}$$

The system is locally weakly observable at $\boldsymbol{x}_0$ if $\boldsymbol{O}$ has full column rank at $\boldsymbol{x}_0$. As the state and measurement equations are infinitely smooth, the observability matrix can have infinite number of rows. If sufficient number of rows which are linearly independent can be found, $\boldsymbol{O}$ is full rank and the system is locally weakly observable.

Based on the observability rank condition, there are a number of publications which show that the platform pose in global frame is unobservable and the rotation around gravity vector (yaw) is unobservable. Some other states are observable, which include the platform pose in the initial camera frame, the gravity vector in the initial camera frame and the features in the camera frame. The intuitive interpretation is that the visual camera is a bearing only sensor, and the IMU is only a double integrator of pose, which are not able to provide the pose and yaw information in the global frame.

The observability analysis allows us to find ways to sufficiently excite the system, such as using three non co-linear features with given world coordinates, or pseudo-measurement equations to make some unobservables observable.[1] Further, it has the potential to provide clues on how to make the unobservables blow-up as slowly as possible.

For filtering-based approaches, the linearised system is employed in EKF. However, the observability of linearised system is different to the corresponding non-linear system. More specifically, the yaw is observable in the linearised systems. This unexpected observable DOF leads to an inconsistent result: the estimated covariance is better than actual measurement result. One direction for improving the consistency is to impose a constraint in the erroneously observable direction due to model linearisation.[38]

The observability was also analysed using a concept of continuous symmetries, which is able to find the analytical derivation of observable state and identifiable parameters.[39]

5.2. Parameter identifiability

In most of visual related localisation techniques, the camera intrinsic parameters are known in advance. They can also be calibrated on-line, such as the methods introduced in [40].

The IMU biases vary with time, and are modelled as time-variant states in most cases. The observability rank condition shows they are observable.

The spatial parameters between IMU and camera are six DOF transformation and must be known precisely to stabilise the estimation results. They are modelled as random walk processes and treated as time-variant states. The analysis of observability rank condition shows that six DOF camera IMU transformation, along with IMU biases, gravity vector and the metric scene structure are all observable. Full observability requires sufficient excitation, i.e. the platform at least undergoes rotation and acceleration along two IMU axes.[41–43]

The temporal parameter between inertial and visual sensor measurements is the time delay. The local identifiability of time delay is established by constructing constraint equations which involve the time delay and other quantities. The rank of the Jacobian matrices of the constraints

is checked for local identifiability. The result shows that the time delay is locally identifiable in general trajectories. The critical trajectories that cause loss of identifiability are characterised.[44] The time delay between inertial and visual sensor measurements is also estimated by a variant of ICP algorithm in [45].

6. Conclusions

This paper has presented an overview of the state-of-the art visual inertial odometry methods. It was presented from two perspectives: filtering-based and optimisation-based approaches. It was also described by handling visual images with two ways: feature based and dense based. The broad knowledge of EKF and image alignment has been unified into the same framework. The links between two approaches have been characterised by iterated EKF, smoother, and marginalisation, and a deep insight into two approaches has been unveiled. The state observability and parameter identifiability are analysed. The paper has also summarised a range of techniques used in this research area, including EKF, MAP, IEKF, BA, ICP, MHE, etc. In the future, exploring the computational complexity of these variants would provide more intuitive guide to the practitioners in this area. More implementation details should be provided as an integrated part of this work.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The first and third author has been financially supported by scholarship from China Scholarship Council.

Notes on contributors

Jianjun Gui received the BSc degree in simulation engineering from the National University of Defense Technology, Changsha, China in 2011 and continued the MSc degree in control science and engineering until 2013. He is currently pursuing the PhD degree in robotics at the University of Essex, Colchester, UK. His research interests include computer vision, pattern recognition, visual inertial odometry, robot pose estimation and visual servoing.

Dongbing Gu received the BSc and MSc degrees in control engineering from the Beijing Institute of Technology, Beijing, China, and the PhD degree in robotics from the University of Essex, Essex, UK. He was an academic visiting scholar with the Department of Engineering Science, University of Oxford, Oxford, UK, from October 1996 to October 1997. In 2000, he joined the University of Essex as a Lecturer. Currently, he is a professor in the School of Computer Science and Electronic Engineering, University of Essex. His current research interests include multiagent systems, wireless sensor networks, distributed control algorithms, distributed information fusion, cooperative control, reinforcement learning, fuzzy logic and neural network-based motion control, and model predictive control.

Sen Wang received the BEng degree in Automation from the Guangdong University of Technology, Guangzhou, China in 2009 and the MEng degree in Control Science and Engineering from Harbin Institute of Technology, Harbin, China, in 2011. He is currently pursuing the PhD degree in robotics at the University of Essex, Colchester, UK. His research interests include robot localisation, SLAM and multiple sensor fusion.

Huosheng Hu received the MSc degree in industrial automation from Central South University, Changsha, China, in 1982 and the PhD degree in robotics from the University of Oxford, Oxford, UK, in 1993. He is currently a professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK, leading the Human-Centred Robotics Group. His research interests include autonomous robots, human–robot interaction, multi-robot collaboration, embedded systems, pervasive computing, sensor integration, intelligent control, cognitive robotics and networked robots. He has published more than 370 research articles in journals, books and conference proceedings. He is a fellow of Institute of Engineering & Technology and Institution of Measurement & Control in the UK, a senior member of IEEE and ACM and a chartered engineer. He is currently an editor-in-chief for the International Journal of Automation and Computing, founding editor-in-chief for Robotics Journal and an executive editor for International Journal of Mechatronics and Automation.

References

[1] Chiuso A, Favaro P, Jin H, et al. Structure from motion causally integrated over time. IEEE Trans. Pattern Anal. Mach. Intell. 2002;24:523–535.
[2] Wolf H. Odometry and insect navigation. J. Exp. Biol. 2011;214:1629–1641.
[3] Corke P, Lobo J, Dias J. An introduction to inertial and visual sensing. Int. J. Rob. Res. 2007;26:519–535.
[4] Franceschini N, Pichon J-M, Blanes C, et al. From insect vision to robot vision [and discussion]. Philos. Trans. R. Soc. B: Biol. Sci. 1992;337:283–294.
[5] Scaramuzza D, Fraundorfer F. Visual odometry [tutorial]. IEEE Rob. Autom. Mag. 2011;18:80–92.
[6] Baker S, Matthews I. Lucas-Kanade 20 years on: a unifying framework. Int. J. Comput. Vision. 2004;56:221–255.
[7] Triggs B, McLauchlan PF, Hartley RI, et al. Bundle adjustment – a modern synthesis. In: Vision algorithms: theory and practice. Berlin:Springer; 2000. p. 298–372.
[8] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I. IEEE Rob. Autom. Mag. 2006;13:99–110.
[9] Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): part II. IEEE Rob. Autom. Mag. 2006;13:108–117.

[10] Dissanayake MG, Newman P, Clark S, et al. A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. Rob. Autom. 2001;17:229–241.

[11] Li M, Mourikis AI. High-precision, consistent EKF-based visual-inertial odometry. Int. J. Rob. Res. 2013;32:690–711.

[12] Li M, Mourikis AI. Improving the accuracy of EKF-based visual-inertial odometry. In: IEEE International Conference on Robotics and Automation (ICRA) Saint Paul, MN, USA; 2012. p. 828–835.

[13] Weiss S, Achtelik MW, Lynen S, et al. Monocular vision for long-term micro aerial vehicle state estimation: a compendium. J. Field Rob. 2013;30:803–831.

[14] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. In: 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR); Nara, Japan 2007. p. 225–234.

[15] Newcombe RA, Lovegrove SJ, Davison AJ. DTAM: dense tracking and mapping in real-time. In: IEEE International Conference on Computer Vision (ICCV) Barcelona, Spain; 2011. p. 2320–2327.

[16] Henry P, Krainin M, Herbst E, et al. RGB-D mapping: using Kinect-style depth cameras for dense 3D modeling of indoor environments. Int. J. Rob. Res. 2012;31:647–663.

[17] Engel J, Schöps T, Cremers D. LSD-SLAM: large-scale direct monocular SLAM. In: European Conference on Computer Vision (ECCV) Zurich, Switzerland; 2014. p. 834–849.

[18] Mourikis AI, Roumeliotis SI. A multi-state constraint Kalman filter for vision-aided inertial navigation. In: IEEE International Conference on Robotics and Automation (ICRA) Rome, Italy; 2007. p. 3565–3572.

[19] Leutenegger S, Furgale PT, Rabaud V, et al. Keyframe-based visual-inertial SLAM using nonlinear optimization [Internet]. In: Robotics: science and systems VI; 2013. Available from http://www.roboticsproceedings.org/rss06/index.html

[20] Strasdat H, Montiel J, Davison AJ. Scale drift-aware large scale monocular SLAM. Rob.: Sci. Syst. 2010;2:5.

[21] Comport AI, Malis E, Rives P. Accurate quadrifocal tracking for robust 3D visual odometry. In: IEEE International Conference on Robotics and Automation (ICRA) Rome, Italy; 2007. p. 40–45.

[22] Steinbrucker F, Sturm J, Cremers D. Real-time visual odometry from dense RGB-D images. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops) Barcelona, Spain; 2011. p. 719–722.

[23] Audras C, Comport A, Meilland M, Rives P. Real-time dense appearance-based SLAM for RGB-D sensors. In: Australasian Conference on Robotics and Automation Melbourne, Australia; 2011.

[24] Tykkala T, Audras C, Comport AI. Direct iterative closest point for real-time visual odometry. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops) Barcelona, Spain; 2011. p. 2050–2056.

[25] Besl PJ, McKay ND. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1992;14:239–256.

[26] Dryanovski I, Valenti RG, Xiao J. Fast visual odometry and mapping from RGB-D data. In: IEEE International Conference on Robotics and Automation (ICRA) Karlsruhe, Germany; 2013. p. 2305–2310.

[27] Newcombe RA, Izadi S, Hilliges O, et al. KinectFusion: real-time dense surface mapping and tracking. In: 10th IEEE International symposium on Mixed and augmented reality (ISMAR) Basel, Switzerland; 2011. p. 127–136.

[28] Pizzoli M, Forster C, Scaramuzza D. REMODE: probabilistic, monocular dense reconstruction in real time. In: IEEE International Conference on Robotics and Automation (ICRA) Hong Kong, China; 2014. p. 2609–2616.

[29] Engel J, Sturm J, Cremers D. Semi-dense visual odometry for a monocular camera. In: IEEE International Conference on Computer Vision (ICCV) Sydney, Australia; 2013. p. 1449–1456.

[30] Forster C, Pizzoli M, Scaramuzza D. SVO: fast semi-direct monocular visual odometry. In: IEEE International Conference on Robotics and Automation (ICRA). Hong Kong, China; 2014. p. 15–22.

[31] Weiss S, Scaramuzza D, Siegwart R. Monocular-SLAM – based navigation for autonomous micro helicopters in GPS-denied environments. J. Field Rob. 2011;28:854–874.

[32] Bell BM, Cathey FW. The iterated Kalman filter update as a Gauss–Newton method. IEEE Trans. Autom. Control. 1993;38:294–297.

[33] Kaess M, Ranganathan A, Dellaert F. ISAM: incremental smoothing and mapping. IEEE Trans. Rob. 2008;24:1365–1378.

[34] Strasdat H, Montiel JM, Davison AJ. Visual SLAM: why filter? Image Vision Comput. 2012;30:65–77.

[35] Sibley G, Matthies L, Sukhatme G. A sliding window filter for incremental SLAM. In: Unifying perspectives in computational and robot vision; Berlin: Springer, 2008. p. 103–112.

[36] Hermann R, Krener AJ. Nonlinear controllability and observability. IEEE Trans. Autom. Control. 1977;22:728–740.

[37] Sontag ED. A concept of local observability. Syst. Control Lett. 1984;5:41–47.

[38] Guo CX, Roumeliotis SI. IMU-RGBD camera 3D pose estimation and extrinsic calibration: Observability analysis and consistency improvement. In: IEEE International Conference on Robotics and Automation (ICRA) Karlsruhe, Germany; 2013. p. 2935–2942.

[39] Martinelli A. Visual-inertial structure from motion: observability and resolvability. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Tokyo, Japan; 2013. p. 4235–4242.

[40] Civera J, Bueno DR, Davison AJ, et al. Camera self-calibration for sequential Bayesian structure from motion. In: IEEE International Conference on Robotics and Automation (ICRA). Kobe, Japan; 2009. p. 403–408.

[41] Kelly J, Sukhatme GS. Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration. Int. J. Rob. Res. 2011;30:56–79.

[42] Mirzaei FM, Roumeliotis SI. A Kalman filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation. IEEE Trans. Rob. 2008;24:1143–1156.

[43] Jones ES, Soatto S. Visual-inertial navigation, mapping and localization: a scalable real-time causal approach. Int. J. Rob. Res. 2011;30:407–430.

[44] Li M, Mourikis AI. Online temporal calibration for camera-IMU systems: theory and algorithms. Int. J. Rob. Res. 2014;33:947–964.

[45] Kelly J, Sukhatme GS. A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors. In: Experimental robotics. Berlin:Springer; 2014. p. 195–209.