DC-VINS: Dynamic Camera Visual Inertial Navigation System with Online Calibration

Jason Rebello Chunshang Li Steven L. Waslander
University of Toronto (UTIAS)
{jason.rebello, chunshang.li}@robotics.utias.utoronto.ca
stevenw@utias.utoronto.ca

Abstract

Visual-inertial (VI) sensor combinations are becoming ubiquitous in a variety of autonomous driving and aerial navigation applications due to their low cost, limited power consumption and complementary sensing capabilities. However, current VI sensor configurations assume a static rigid transformation between the camera and IMU, precluding manipulating the viewpoint of the camera independent of IMU movement which is important in situations with uneven feature distribution and for high-rate dynamic motions. Gimbal stabilized cameras, as seen on most commercially available drones, have seen limited use in SLAM due to the inability to resolve the time-varying extrinsic calibration between the IMU and camera needed in tight sensor fusion. In this paper, we present the online extrinsic calibration between a dynamic camera mounted to an actuated mechanism and an IMU mounted to the body of the vehicle integrated into a Visual Odometry pipeline. In addition, we provide a degeneracy analysis of the calibration parameters leading to a novel parameterization of the actuated mechanism used in the calibration. We build our calibration into the VINS-Fusion package and show that we are able to accurately recover the calibration parameters online while manipulating the viewpoint of the camera to feature rich areas thereby achieving an average RMSE error of 0.26m over an average trajectory length of 340m, 31.45% lower than a traditional visual inertial pipeline with a static camera.

1. Introduction

The ability of a robot to perform accurate Simultaneous Localization and Mapping (SLAM) in an unknown environment depends on the information perceived from its surroundings. Although SLAM has been extensively studied over the past few decades, more recently it has entered an era focused on robust performance, high-level understanding, resource awareness and task-driven perception [3]. This is particularly important when dealing with aerial vehicles such as drones with severe payload limitations and computational constraints. Visual-inertial (VI) sensor configurations offer complementary properties which make them particularly suitable in applications where efficient, robust and accurate SLAM is key.

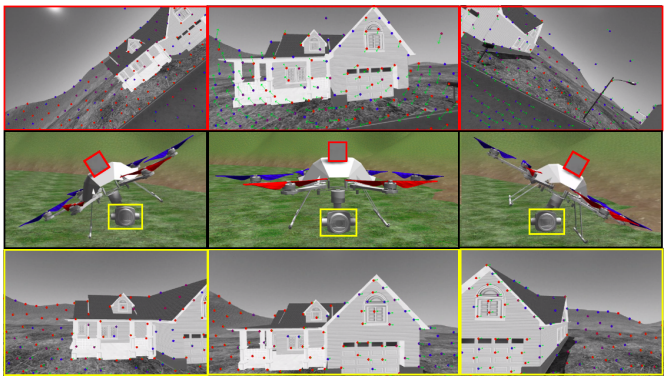


Figure 1: Static (red) and gimbal-stabilized dynamic camera (yellow) images on an aerial vehicle while performing aggressive motions.

While visual sensors provide rich high-dimensional data capable of capturing detailed appearance information and performing accurate long-term localization, these sensors are sensitive to situations involving motion blur, occlusions and illumination changes typically encountered in aerial applications. On the other hand, IMUs provide high frequency accelerometer and gyroscopic measurements which when integrated provide accurate short-term pose estimates in high dynamic motion profiles. Current visual-inertial sensor configurations [24, 22] assume a rigid transformation between the two sensors, thereby coupling the viewpoint of the camera to the motion of the IMU. As a result, the cameras in these systems can experience motion blur and reduced performance in feature initialization and tracking.

Although sensors such as Event Cameras can be used for fast motion tracking [25], camera *gimbals* present on most commercially available drones, most commonly for image stabilization and videography viewpoint management can be used for the same. In the VI setting, the gimbal can significantly reduce motion blur and allow for smoother image transitions irrespective of vehicle motion, leading to more accurate pose estimates [26, 40]. Fig. 1 shows the difference between image viewpoints in simulation for a static camera rigidly mounted to the vehicle and a gimbal stabilized camera while performing aggressive dynamic motions. However, the use of gimballed cameras in tightly coupled visual-inertial SLAM applications has not been previously demonstrated due to the inability to reliably resolve the time-varying extrinsic calibration between the camera and the IMU through the actuated mechanism.

Errors in the extrinsic calibration between the dynamic camera and IMU appear as measurement biases and reduce the overall accuracy of visual inertial state estimation. While the combination of an offline IMU to static camera calibration [11] and an extrinsic calibration from static camera to dynamic camera can be employed [33], these methods are time-consuming and can only be performed in situations where a calibration target is available. In contrast, online calibration methods have the ability to re-calibrate the sensor configuration on the fly and handle changes caused by wear, sensor re-positioning, or mechanical stress.

In this paper, we develop an online extrinsic calibration between a dynamic camera and IMU. We build our calibration into the VINS-Fusion [29] package and show that our method is capable of estimating the calibration online and in flight. To the best of our knowledge, this is the first work to recover the extrinsic calibration between a dynamic camera and an IMU through an actuated mechanism. We test our method in the RotorS simulator [12] and improve RMSE as compared to the combined offline calibration. We also performs tests to demonstrate the utility of dynamic cameras in high speed, dynamic aerial applications performing visual inertial navigation. In addition to the novel online calibration approach, we identify calibration parameters that cause the system to enter a degenerate state when joint angle values are not available, leading to a non-unique solution for the calibration. We propose a novel parameterization of the actuated mechanism, leading to fewer calibration parameters while crucially resolving the degeneracy, leading to a more accurate and repeatable calibration.

2. Related Works

2.1. Kinematic and Dynamic Camera Calibration

The calibration relating dynamic cameras and manipulators are closely related to other calibrations such as hand-in-eye [6], head-to-eye [21] and kinematic calibration [27, 9]. Although initially developed for camera to end-effector calibration, [4] applied this method to the calibration of a dynamic camera and the odometry frame of the vehicle. [41] provide global optimality guarantees on the recovered calibration parameters. Recently, [23] performed kinematic calibration using an RGBD camera while mapping and localizing in an unknown environment.

A separate body of work deals with the calibration of Dynamic Camera Clusters (DCCs) [8, 32], where they seek to recover the transformation from a camera mounted to an actuated mechanism to another camera that is rigidly mounted to the vehicle. In [5], unknown encoder angles were added to the offline calibration procedure and later tested in a Visual Odometry application. Recently, [33] extended the calibration to use a pose-loop formulation as opposed to a pixel error formulation to achieve better measurement excitation while providing an analysis of the degenerate parameters when joint angle values are not available. While previous methods used a fiducial target to resolve the calibration, our method recovers the transformation from a dynamic camera to an IMU online and in flight while relying solely on natural features in the environment.

2.2. VINS systems

Visual inertial navigation systems (VINS) can broadly be divided into optimization and filtering algorithms. While filtering-based VINS [24, 1, 16] have demonstrated high-accuracy state estimation, they suffer from the limitation of a one-time linearization which can degrade performance especially in systems with non-linear measurement functions. Batch optimization methods [22, 29, 39], can achieve higher accuracy by solving the bundle adjustment problem over a set of measurements, allowing the error to reduce through re-linearization while incurring higher computational cost. There is also a large body of research on recovering the temporal and spatial calibration between an IMU and camera [10, 34, 31]. However, these systems all assume a fixed calibration between the IMU and camera.

2.3. Extrinsic Calibration using Neural Networks

With the advancement of deep learning several methods have tried to address the problem of extrinsic calibration between various types of sensors [43, 36]. One of the first methods to employ neural networks for the extrinsic spatial calibration between a lidar and a camera was demonstrated in RegNet [35], where they use a series of convolutional neural networks followed by network-in-network blocks to resolve the difference between the predicted and ground truth calibrations. Later CalibNet [17], performed the same calibration by minimizing the photometric and geometric error between the input images and 3D point clouds. We discuss in Sec. 4.1, the inability to use deep learning methods and state the advantage of using traditional calibration

algorithms to resolve our dynamic calibration.

2.4. Degeneracy Analysis

The analysis of degenerate configurations has been well studied with several works dealing with multi-camera [38, 19, 20, 33] and visual inertial [18, 42] systems. Degeneracy analysis of a system is crucial to understanding if the required estimation parameters can be uniquely recovered using the available observations [15]. For non-linear systems that do not possess dynamics, determining the degenerate parameters is equivalent to identifying the columns of the measurement jacobian that cause it to be rank deficient. [38] successfully identify degenerate motions for non-overlapping multi-camera systems in visual SLAM applications, both from geometric and non-linear optimization techniques respectively. [18] and [42] analyze the VI sensor configuration motions that cause the system to be degenerate for navigation with spatial and temporal calibration.

3. BACKGROUND AND NOTATION

Frames and Notation: Let a point in 3D, expressed in co-ordinate frame $\mathcal{F}_x$, be denoted as $\mathbf{p}^x \in \mathbb{R}^3$. We define a rigid body transformation from frame $\mathcal{F}_a$ to $\mathcal{F}_b$ as $\mathbf{T}_\tau^{b:a} \in \mathbb{SE}(3)$, parameterized using a 6-DOF vector τ and is made up of a rotation $\mathbf{R}^{b:a} \in \mathbb{SO}(3)$ and translation $\mathbf{t}^{b:a} \in \mathbb{R}^3$ between the frames and is expressed in matrix form as,

$$\mathbf{T}_\tau^{b:a} = \begin{bmatrix} \mathbf{R}^{b:a} & \mathbf{t}^{b:a} \\ 0 & 1 \end{bmatrix} \tag{1}$$

Projection Model and PnP Solution: We define a projection model $\Psi(\mathbf{p}^c) : \mathbb{R}^3 \mapsto \mathbb{P}^2$ that maps a point expressed in camera frame, $\mathcal{F}_c$, to a pixel location on the 2D image plane. Given a set of known 3D points in a co-ordinate frame $\mathcal{F}_w$ and its corresponding pixel location on the image plane, we can resolve the true pose of the camera in the frame $\mathcal{F}_w$ via the Perspective-n-Point (PnP) solution.

Denavit-Hartenberg Parameterization: We make use of the well established Denavit-Hartenberg (DH) convention [14] to parameterize the kinematic chain which incorporates 4 independent parameters $\omega_l = [\, d_l,\, a_l,\, \alpha_l]^T$ and θ_l, where $\theta_l, \alpha_l \in [0, 2\pi)$ and $d_l, a_l \in \mathbb{R}$. The time-varying parameter in this representation is the joint angle parameter θ_l, with the remaining parameters being static. With successive co-ordinate frames defined for each of the joints, the transformation from frame $\mathcal{F}_i$ to $\mathcal{F}_{i-1}$ can be computed as follows:

$$\mathbf{T}_{\omega_l,\theta_l}^{i-1:i} = \begin{bmatrix} \cos\theta_l & -\sin\theta_l\cos\alpha_l & \sin\theta_l\sin\alpha_l & a_l\cos\theta_l \\ \sin\theta_l & \cos\theta_l\cos\alpha_l & -\cos\theta_l\sin\alpha_i & a_l\sin\theta_l \\ 0 & \sin\alpha_l & \cos\alpha_l & d_l \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Degeneracy Analysis: While the optimization of transformation matrices can be performed over $\mathbb{SE}(3)$, we perform the optimization and degeneracy analysis by treating the rotation part of the transformation matrix as a manifold in $\mathbb{SO}(3)$, but using the translation component as a vector space in $\mathbb{R}^3$, similar to the analysis in [7]. This enables the use of the derivatives in [2] which leads to easier analysis of the degeneracies and the ability to rely on identities Eq.(3.17), Eq.(3.18) and Eq.(3.24) in [7], omitted due to space considerations. We let I be a 3x3 identity matrix and $[\mathbf{A}]_i$ represent the i^{th} column of the matrix $\mathbf{A}$. We denote $[\cdot]^\wedge$ as the transformation of a vector to a skew symmetric matrix, while $[\cdot]^\vee$ denotes its inverse operation. We make use of the following two identities for any rotation matrix $\mathbf{R}$ and any vector $\mathbf{v}$ [2]:

$$[\mathbf{R}\mathbf{v}]^\wedge = \mathbf{R}[\mathbf{v}]^\wedge\mathbf{R}^T \quad (3) \qquad\qquad \mathbf{R}[\mathbf{v}]^\wedge = [\mathbf{R}\mathbf{v}]^\wedge\mathbf{R} \quad (4)$$

4. Problem Formulation

4.1. Classical vs Deep Calibration

In this section we briefly describe the inability to use deep learning methods to resolve our dynamic calibration and state the importance of approaching this problem using a classical formulation. In this paper we seek to recover a time-varying extrinsic transformation between the camera and IMU that is governed by a collection of fixed (ω, τ) and dynamic (θ) parameters used to define the transformation. The relative transformation between the camera and IMU changes with each change in joint angle value. While one could fix a particular configuration of joint angle values and establish a fixed 6-DOF transformation between the camera and IMU using a package like Kalibr, collecting such a dataset for every possible joint angle configuration would not only be time consuming and cumbersome but would lead to poor generalizability with the increase in number of joints as well as possible gimbal configurations (order of joints such as yaw-roll-pitch, pitch-yaw-roll etc.). This would also require the presence of encoders on the gimbal (typically not available on most commercial drones) to ensure the same joint angles are achieved. At the same time, the ability to recover a unique calibration is contingent on identifying the parameters that cause the calibration to become degenerate. To the best of our knowledge, we are unaware of any deep learning method capable of providing this information. We therefore resort to traditional approaches of manipulator kinematics and pixel-error formulations to resolve our calibration.

4.2. Manipulator Chain Description:

In this section we first provide an in-depth description of the parameterization used to describe the transformation

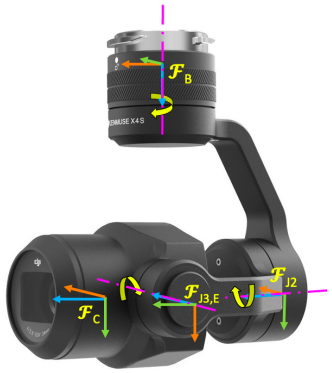


Figure 2: DJI Zenmuse X4S 3-axis gimbal with a camera.

from the dynamic camera frame $\mathcal{F}_C$, to the IMU frame, $\mathcal{F}_I$. It should be noted, that while we use a 3-DOF gimbal as shown in Fig. 2, our formulation is applicable to a mechanism with any number of joints. The transformation from the dynamic camera to IMU is given by

$$\mathbf{T}^{I:C}_{\Phi,\boldsymbol{\theta}} = \mathbf{T}^{I:B}_{\tau_I}\mathbf{T}^{B:E}_{\boldsymbol{\omega},\boldsymbol{\theta}}\mathbf{T}^{E:C}_{\tau_C} \tag{5}$$

where $\Phi = \{\tau_I, \boldsymbol{\omega}, \tau_C\}$ are a collection of all the static parameters describing the transformation of the actuated chain. $\mathbf{T}^{I:B}_{\tau_I}$ and $\mathbf{T}^{E:C}_{\tau_C}$ are rigid 6-DOF static transformations from the base frame of the mechanism, $\mathcal{F}_B$ to the IMU frame and from the dynamic camera frame to the end-effector frame $\mathcal{F}_E$ respectively. $\mathbf{T}^{B:E}_{\omega,\theta}$ is a series of transformations relating information through the (N=3-DOF) actuated gimbal and is given as

$$\mathbf{T}^{B:E}_{\boldsymbol{\omega},\boldsymbol{\theta}} = \mathbf{T}^{B:J2}_{\omega_1,\theta_1}\mathbf{T}^{J2:J3}_{\omega_2,\theta_2}\mathbf{T}^{J3:E}_{\omega_3,\theta_3} \tag{6}$$

where $\boldsymbol{\omega} = \{\omega_1, \omega_2, \omega_3\}$ and $\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3\}$ are a collection of the static DH parameters and joint angles respectively.

4.3. DC-VINS State Vector Formulation

VINS-Fusion is capable of recovering a static 6-DOF extrinsic transformation between a camera and an IMU mounted to the vehicle in flight. Once calibrated, this transformation does not change with time. However, when using a dynamic camera in visual inertial applications, one needs to estimate a series of transformations as described in Eq. (5). We therefore extend the VINS-Fusion package to now estimate all the calibration parameters Φ as well as the joint angles $\boldsymbol{\theta}$ at each time-step. For a complete description of the original VINS-Fusion method, we refer the reader to [30, 28].

Let us assume a sliding window of size n consisting of IMU states as well as m features observed by the keyframes in this window. The full state vector, $\mathcal{X}$, can be defined as:

$$\begin{aligned}
\mathcal{X} &= [\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_n, \lambda_1, \lambda_2, ..., \lambda_m, \Phi, \Theta] \\
\mathbf{x}_k &= [\mathbf{p}^w_{I_k}, \mathbf{v}^w_{I_k}, \mathbf{q}^w_{I_k}, \mathbf{b}_a, \mathbf{b}_g], k \in [0, n] \\
\Phi &= [\tau_I, \boldsymbol{\omega}, \tau_C] \\
\Theta &= [\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n]
\end{aligned} \tag{7}$$

where $\mathbf{x}_k$ is the state of the IMU when the k^{th} image is taken and consists of the position ($\mathbf{p}^w_{I_k}$), velocity ($\mathbf{v}^w_{I_k}$) and orientation ($\mathbf{q}^w_{I_k}$) of the IMU in the world frame as well as the accelerometer ($\mathbf{b}_a$) and gyroscope bias ($\mathbf{b}_g$) in the IMU body frame. λ_l is the inverse depth of the l^{th} feature from the first observation in the respective camera frame. Φ is the collection of static parameters as described in Sec. 4.2 and is constant across all the frames in the window. In order to account for the different dynamic camera viewpoints at the different time-stamps, we need to estimate the joint angles for each frame in the window. Let the collection of all the joint angles needed to be estimated be represented as Θ, where each $\boldsymbol{\theta}_i$ is a set of L angles for an L-joint mechanism.

Online calibration of the DC-VINS system involves signficantly more parameters than the six parameters in the static case. For an actuated mechanism with L links we have a total of $12 + 3L + Ln$ parameters to be estimated with 12 parameters from the two 6-DOF rigid transformations $\mathbf{T}^{I:B}_{\tau_I}$ and $\mathbf{T}^{E:C}_{\tau_C}$, $3L$ parameters for the static DH parameters, ω_l, for each of the L links, and Ln parameters for the L joint angles for a n-window system. Therefore, for a window size of 10 and a 3 joint mechanism we have $12 + 3 \cdot 3 + 3 \cdot 10 = 51$ parameters. Despite an expected reduction in calibration accuracy for the actuated chain over a static transformation, we show that a net gain in VI estimation performance can be obtained when an actuated camera is able to stabilize image capture and diminish dynamic motion effects.

Visual-inertial bundle adjustment is then formulated by minimizing the sum of the prior and the Mahalanobis distance of all visual and IMU measurement residuals in the sliding window in order to obtain a posterior estimate as follows,

$$\begin{aligned}
\min_{\mathcal{X}} \Bigg\{ \left\| \mathbf{r}_p - \mathbf{H}_p\mathcal{X} \right\|^2 + \sum_{k\in\mathcal{I}} \left\| \mathbf{r}_{\mathcal{B}}\left(\hat{\mathbf{z}}^{I_k}_{I_{k+1}}, \mathcal{X}\right) \right\|^2_{\mathbf{P}^{I_k}_{I_{k+1}}} \\
+ \sum_{(l,j)\in\mathcal{C}} \rho\left(\left\| \mathbf{r}_{\mathcal{C}}\left(\hat{\mathbf{z}}^{c_j}_l, \mathcal{X}\right) \right\|^2_{\mathbf{P}^{c_j}_l} \right) \Bigg\}
\end{aligned} \tag{8}$$

where $\mathbf{r}_p$ and $\mathbf{H}_p$ are the prior information from the previous marginalization step, $\mathbf{r}_{\mathcal{B}}\left(\hat{\mathbf{z}}^{I_k}_{I_{k+1}}, \mathcal{X}\right)$ is the IMU residual term as defined in [29] and $\mathbf{r}_{\mathcal{C}}\left(\hat{\mathbf{z}}^{c_j}_l, \mathcal{X}\right)$ is the visual residual term as described in Eq. (9) below, operated on by

the Huber norm ρ. $\mathbf{P}_{I_{k+1}}^{I_k}$ and $\mathbf{P}_l^{c_j}$ are the measurement covariance for the IMU and visual terms respectively. $\mathcal{I}$ is the total number of IMU measurements and $\mathcal{C}$ is the total number of features that are observed in at least two images in the sliding window.

4.4. DC-VINS Visual Measurement Residual

The visual error term is formulated by considering the camera and IMU instances across two time-steps. Since the original VINS-Fusion consists of a static transformation between the camera and IMU, this transformation remains constant across all time-steps. Our formulation on the other hand allows the movement of the camera between two time-steps, therefore having different transformations from the dynamic camera to IMU at each instance. Let the transformation from the dynamic camera to IMU at time-steps i and j be represented as $\mathbf{T}_{\Phi,\boldsymbol{\theta}_i}^{I_i:C_i}$ and $\mathbf{T}_{\Phi,\boldsymbol{\theta}_j}^{I_j:C_j}$. These respective transformations are goverened by the set of joint angles $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ applied respectively to the mechanism.

If we consider the l^{th} feature first observed in the i^{th} camera frame, the pixel-error residual of the feature observation in the j^{th} camera frame is given by

$$\begin{aligned} \mathbf{r}_{\mathcal{C}}\left(\hat{\mathbf{z}}_l^{c_j}, \mathcal{X}\right) &= \mathbf{z}_l^{c_j} - \hat{\mathbf{z}}_l^{c_j} \\ \mathbf{z}_l^{c_j} &= \Psi(\mathbf{p}^{c_j}) \end{aligned} \tag{9}$$

where $\hat{\mathbf{z}}_l^{c_j}$ is the true pixel measurement in the j^{th} image frame while $\mathbf{z}_l^{c_j}$ is the projected pixel after transforming the 3D point, $\mathbf{p}^{c_i}$ expressed in the i^{th} image frame through the actuated mechanism and is given by,

$$\mathbf{p}^{c_j} = \underbrace{(\mathbf{T}_{\Phi,\boldsymbol{\theta}_j}^{I_j:C_j})^{-1}(\mathbf{T}_{\tau_j}^{W:I_j})^{-1}(\mathbf{T}_{\tau_i}^{W:I_i})(\mathbf{T}_{\Phi,\boldsymbol{\theta}_i}^{I_i:C_i})}_{\mathbf{T}_{,}^{C_j:C_i}} \mathbf{p}^{c_i} \tag{10}$$

where $\mathbf{p}^{c_i} = \frac{1}{\lambda_l}\Psi^{-1}\left(\begin{bmatrix} \hat{u}_l^{c_i} & \hat{v}_l^{c_i} \end{bmatrix}^T\right)$ is the back projection function, which turns the first pixel observation $\hat{u}_l^{c_i}, \hat{v}_l^{c_i}$ along with the inverse depth λ_l into a 3D point.

4.5. Degeneracy Analysis

In this section we identify the degenerate parameters of the visual-inertial calibration problem that cause the system to go to into an irrecoverable state leading to non-uniqueness of the calibration parameters. It should be noted that the degeneracy presented here is a function of the parameterization of the actuated chain, therefore existing for any angle configuration of the mechanism. While a similar analysis was presented in [7] and [33], the identification was made for a single gimbal chain incorporating two cameras while our analysis uses a double chain formulation as described in Eq. (10). Due to space restrictions, we limit our analysis to a single set of parameters, in particular the degeneracy arising between the translation parameters describing the Base to IMU transformation, $\mathbf{T}^{I:B}$ and the d parameter describing the transformation $\mathbf{T}^{B:J2}$. We refer the reader to [7, 33] for a similar analysis of the remaining parameters. The Jacobian of Eq. (9) with respect to $\mathbf{T}^{I:B}$ is given by,

$$\frac{\partial \mathbf{r}_{\mathcal{C}}}{\partial \mathbf{T}^{I:B}} = \underbrace{\frac{\partial \mathbf{r}_{\mathcal{C}}}{\partial \mathbf{z}_l^{c_j}} \frac{\partial \mathbf{z}_l^{c_j}}{\partial \mathbf{p}^{c_j}} \frac{\partial \mathbf{p}^{c_j}}{\partial \mathbf{T}^{C_j:C_i}}}_{\mathcal{J}_1} \left[\underbrace{\frac{\partial \mathbf{T}^{C_j:C_i}}{\partial \mathbf{T}^{C_j:I_j}} \frac{\partial \mathbf{T}^{C_j:I_j}}{\partial \mathbf{T}^{I_j:C_j}}}_{\mathcal{J}_2} \frac{\partial \mathbf{T}^{I_j:C_j}}{\partial \mathbf{T}^{I_j:B_j}} + \underbrace{\frac{\partial \mathbf{T}^{C_j:C_i}}{\partial \mathbf{T}^{I_i:C_i}}}_{\mathcal{J}_3} \frac{\partial \mathbf{T}^{I_i:C_i}}{\partial \mathbf{T}^{I_i:B_i}} \right] \tag{11}$$

We note that while the overall transformation of the actuated chain from dynamic camera to IMU varies at different time-steps, the transformation from the base of the mechanism to the IMU and dynamic camera to end-effector are constant irrespective of time-step. Therefore we have, $\mathbf{T}^{I_i:B_i}$=$\mathbf{T}^{I_j:B_j}$=$\mathbf{T}^{I:B}$, and similarly for $\mathbf{T}^{E:D}$. While determining Jacobian $\mathcal{J}_1$ is trivial, we briefly state the Jacobians $\mathcal{J}_2$ and $\mathcal{J}_3$, where we make use of Eq.(3.18) and Eq.(B.4) from [7],

$$\mathcal{J}_2 = \begin{bmatrix} -\mathbf{R}^{C_j:I_j} & 0_{3\text{x}3} \\ -\mathbf{R}^{C_j:I_j}[\mathbf{t}^{I_j:C_i} - \mathbf{t}^{I_j:C_j}]^\wedge & -\mathbf{R}^{C_j:I_j} \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{12}$$

$$\mathcal{J}_3 = \begin{bmatrix} \mathbf{R}^{C_j:I_i} & 0_{3\text{x}3} \\ 0_{3\text{x}3} & \mathbf{R}^{C_j:I_i} \end{bmatrix} = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \tag{13}$$

Considering a single chain $\mathbf{T}^{I_x:C_x}$, $x \in \{i, j\}$ we state the Jacobian with respect to $\mathbf{T}^{I_x:B_x}$ and refer the reader to [7] for the derivation.

$$\frac{\partial \mathbf{T}^{I_x:C_x}}{\partial \mathbf{T}^{I_x:B_x}} = \begin{bmatrix} I & 0_{3\text{x}3} \\ -[\mathbf{R}^{I_x:B_x}\mathbf{t}^{B_x:C_x}]^\wedge & I \end{bmatrix} \tag{14}$$

where the first 3 and last 3 columns represent the Jacobians with respect to the rotation and translation parameters respectively. Similarly, the Jacobian with respect to the d parameter for $\mathbf{T}^{B_x:J2_x}$, is given by

$$\frac{\partial \mathbf{T}^{I_x:C_x}}{\partial \mathbf{T}_d^{B_x:J2_x}} = \begin{bmatrix} 0_{3\text{x}1} \\ [\mathbf{R}^{I_x:B_x}]_3 \end{bmatrix} \tag{15}$$

Substituting Eq. (12), (13) and the translation Jacobians of Eq. (14) into Eq. (11), we get

$$\frac{\partial \mathbf{r}_{\mathcal{C}}}{\partial \mathbf{T}^{I:B}} = \mathcal{J}_1 \begin{bmatrix} [A_2 + B_2]\, I \end{bmatrix} \tag{16}$$

Similarly we can show that the d Jacobian in Eq. (15) can be written as

$$\frac{\partial \mathbf{r}_{\mathcal{C}}}{\partial \mathbf{T}_d^{B:J2}} = \mathcal{J}_1 \begin{bmatrix} [A_2 + B_2]\, [\mathbf{R}^{I:B}]_3 \end{bmatrix} \tag{17}$$

Since $\mathbf{T}^{I:B}$ is a static transformation, we can multiply Eq. (16) with $[\mathbf{R}^{I:B}]_3$ resulting in Eq. (17), thereby leading to the degeneracy. We can similarly show that there are degeneracies related to the d, a, α and θ parameters of the $\mathbf{T}^{J3:E}$ link and the d and θ parameters of the $\mathbf{T}^{B:J2}$ link as described in [7, 33] leading to a total of 6 degenerate parameters.

4.6. Minimal Parameterization

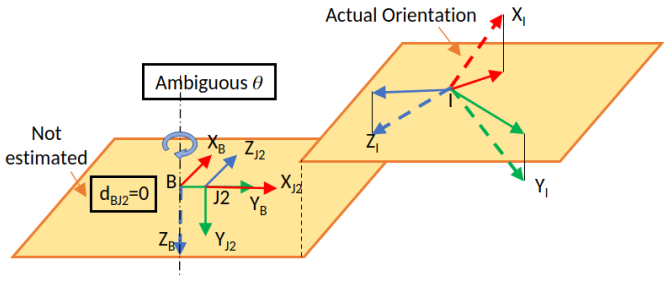
The transformation chain described in Eq. (5) is overparameterized, therefore, in this section we propose a *novel minimal* parameterization eliminating previous degeneracies, which is a crucial requirement in having the ability to recover a unique set of calibration parameters. Assuming a 3-DOF gimbal, the transformation from camera frame to IMU frame for a single configuration consists of two 6-DOF transformations and three DH transformations resulting in a total of 12+3*4=24 parameters. However, as stated in Sec. 4.5, such a parameterization had 6 degenerate parameters resulting in a total of 24-6=18 minimal parameters needed to define the actuated chain from dynamic camera to IMU.

4.6.1 Last Joint Degeneracy Resolution

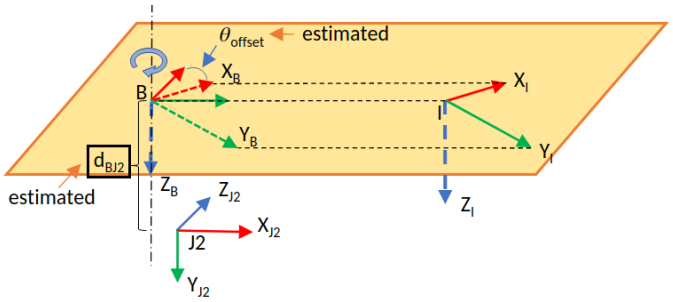
According to the old parameterization, the transformation from the last joint $J3$ to the dynamic camera C is made up of two transformations, a 4-DOF DH transformation $\mathbf{T}^{J3:E}$ and a 6-DOF static transformation $\mathbf{T}^{E:C}$, leading to a total of 10 parameters used to define $\mathbf{T}^{J3:C}$. However, as shown in [7, 33], the DH parameters used to describe the $\mathbf{T}^{J3:E}$ are degenerate leading to a total of 10-4=6 minimal parameters used to describe $\mathbf{T}^{J3:C}$. In order to preserve the time-varying nature of the θ parameter describing $\mathbf{T}^{J3:E}$, we propose to augment the DH parameters by appending two other transformations, a rotation β and a translation y around and along the y-axis respectively in order to achieve an augmented 6-DOF DH transformation from $J3$ to C. The entire minimal transformation for $\mathbf{T}^{J3:C}$ is then given by $\mathbf{T}^{J3:C} =$

$$\mathbf{T}_{DH}^{J3:E} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\beta) & 0 & -cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$= \begin{bmatrix} c_\theta c_\beta - s_\theta s_\alpha s_\beta & -s_\theta c_\alpha & c_\theta s_\beta + s_\theta s_\alpha c_\beta & -y s_\theta c_\alpha + a c_\theta \\ s_\theta c_\beta + c_\theta s_\alpha s_\beta & c_\theta c_\alpha & s_\theta s_\beta - c_\theta s_\alpha c_\beta & y c_\theta c_\alpha + a s_\theta \\ -c_\alpha s_\beta & s_\alpha & c_\alpha c_\beta & y s_\alpha + d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$



(a) Parameterization degeneracy from [8] with unresolved joint angle value



(b) Proposed parameterization including theta offset to resolve degeneracy.

Figure 3: Visual depiction of first joint degeneracy resolution. Note the change in estimation of d_{BJ2} and θ

.

where c_x and s_x, $x \in \{\theta, \alpha, \beta\}$ represent the cos and sin values of the parameters and $\mathbf{T}_{DH}^{J3:E}$ is as defined in Eq. (2). While we note that this parameterization will produce a gimbal lock situation at $\alpha = |90|$, this can be accounted for by switching the rotation around the α and β axis leading to a minimal parameterization of $\mathbf{T}^{J3:C}$.

4.6.2 First Joint Degeneracy Resolution

The transformation from joint $J2$ to the IMU I is made up of two transformations, a 4-DOF DH transformation from joint $J2$ to the base of the mechanism $\mathbf{T}^{B:J2}$ and a 6-DOF transformation from the base to the IMU frame, $\mathbf{T}^{I:B}$. As shown in [7, 33], the d and θ parameters on the $\mathbf{T}^{B:J2}$ are degenerate leading to a total of 10-2=8 parameters used to define the transformation $\mathbf{T}^{I:J2}$. This degeneracy in d arises due to the fact that the only requirement for placing the base co-ordinate frame, is that the origin and z-axis of the base frame should lie along the axis of joint angle rotation leading to an ambiguous absolute position. In the case when joint angle values are not available, the relative rotation between the IMU and joint $J2$ can be captured by both, the joint angle value θ and the relative 3-DOF rotation between the base and IMU leading to an ambiguous rotation resulting in a degeneracy in the θ parameter. In order to eliminate these degeneracies, we propose limiting the transformation

from from base to IMU frame from a 6-DOF to a 4-DOF with two rotations and two translations around the x and y axis respectively. The transformation $\mathbf{T}^{I:J2}$ can then be represented as $\mathbf{T}^{I:J2} =$

$$\begin{bmatrix} c_{ry} & 0 & s_{ry} & t_x c_{ry} \\ s_{rx}s_{ry} & c_{rx} & -c_{ry}s_{rx} & t_y c_{rx} + t_x s_{rx} s_{ry} \\ -c_{rx}s_{ry} & s_{rx} & c_{rx}c_{ry} & t_y s_{rx} - t_x c_{rx} s_{ry} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{T}_{DH}^{B:J2} \tag{20}$$

where c_x and s_x, $x \in \{rx, ry\}$ represent the cos and sin values of the parameters, rx, ry, tx and ty represent the rotations and translations along the x and y axis respectively. As shown in Fig.3, this 4-DOF transformation along with the optimization of all the DH parameters in $\mathbf{T}^{B:J2}$, is capable of recovering any arbitrary transformation from IMU to Base frame, thereby eliminating the degeneracy resulting in a minimal parametreization.

5. Experiments

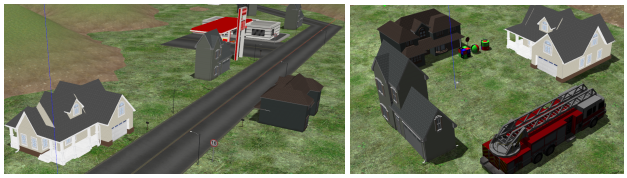
The calibration method in this paper is focused on two main reasons: 1) To perform the calibration between a dynamic camera and IMU online and in flight and 2) To show the advantage of a dynamic camera over a static camera while performing aggressive aerial motions. We compare our method against a *make-shift* offline dynamic camera to IMU calibration and test the calibrations in a simulated environment consisting of Gazebo and the RotorS simulator. We evaluate both calibrations by comparing it to the ground truth values while also noting the difference in the RMSE and mean error over the entire run.

5.1. Simulation Environment

In order to perform our experiments we make use of the RotorS Simulator [12] which is an open-source, customizable MAV gazebo simulator. It supports several high quality aerial robot models as well as a variety of sensors such as cameras and IMUs. We simulate a variety of environments consisting of various buildings that would typically be available in the real world as seen in Fig.4. For our experiments we make use of a firefly drone and a affix a custom built 3-axis gimbal, based on the model shown in Fig.2 to the base of the mechanism as shown in Fig.1. We limit the movement of the gimbal to ± 180, ± 25 and 30 to -120 degrees in the yaw, roll and pitch angles respectively, while using an existing PID controller on each of the joints to perform gimbal stabilization. We also attach a static camera to the drone in order to perform comparisons with static VI calibration.

5.2. Offline Dynamic Camera to IMU calibration

Offline calibration is performed in two steps, by calibrating IMU to static camera with Kalibr [11], and static to dynamic camera via DCC calibration [33]. We employ the OpenVSLAM [37] package on a simulated environment of buildings and roads, collecting 39 pose measurements by moving the drone around in the constructed map and randomly exciting the gimbal over the entire operating workspace.



(a) Environment 1 (b) Environment 2

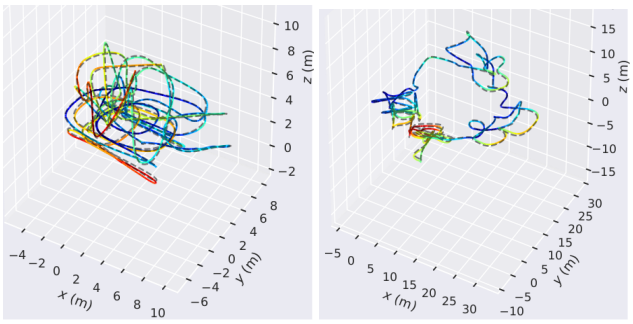
Figure 4: Simulated Gazebo environments used for experiments.

We note that in order to perform a fair comparison of the calibration quality, we collect all calibration data on the firefly drone during flight operation. A total of 2 minutes of camera data was collected for intrinsic calibration, while 3 minutes for the IMU to camera calibration while providing sufficient excitation to drone motion. We note the mean (M) and standard deviation (SD) of the pixel error for both calibration procedures. Since we have the ground truth extrinsic transformation from static camera to IMU, we also report the mean translation and rotational errors along with the standard deviation values in the Table 1.

Average mean and standard deviation pixel error from the pose measurements for the static (DCC static) and gimbal (DCC gimbal) over the entire measurement set are reported in Table 1. After performing DCC calibration (DCC calib.), we achieve a mean pixel error of 1.54 pixels along with a mean translation error of 7.78mm and a rotation error of 4.23 degrees on the calibrated DH values. It is important to note that the we add zero mean noise with a standard deviation of 10 degrees on each of the encoder angles. We also note that greater excitation in measurement collection over the gimbal work-space coupled with lower encoder noise, reduced the rotational error to a mean of 0.31 degrees with a standard deviation of 0.15.

	Avg. pix err		Avg. trans (mm)		Avg. rot (deg)	
	M	Std	M	Std	M	Std
Cam. Int.	0.31	0.12	-	-	-	-
Cam.-IMU	0.36	0.17	12.30	5.02	0.35	0.11
DCC static	1.35	0.12	-	-	-	-
DCC gimbal	1.44	0.13	-	-	-	-
DCC calib.	1.54	0.22	7.78	4.02	4.23	2.71

Table 1: Offline Dynamic Camera to IMU calibration



(a) T1 experiment trajectory (b) T3 experiment trajectory

Figure 5: Example trajectories of in experiments T1 and T3.

5.3. Dynamic versus Static Camera VINS

In order to demonstrate the effectiveness of a gimbal stabilized dynamic camera on a drone performing aggressive aerial motions, we test our method in a visual inertial application using the VINS Fusion package. We perform 3 sets of experiments in two simulated environments as shown in Fig.4 and report the average RMSE, Mean and Standard Deviation error values in Table 2 using the EVO package [13]. We note, that in all the experiments we avoid the trivial case when the static camera does not observe any features due to high speed motions but emphasize that there is significant advantage in having the ability to point the gimbal camera to feature rich areas that aid in the pose estimation quality. In experiment T1, we fly the drone around a house in Environment 1, while performing random aggressive aerial motions as shown in Fig.5(a). As can be seen from the results there is a clear advantage in using the gimbal camera over the static leading to a 23.95% error decrease in the RMSE value. Experiment T2 tests the estimation quality while flying the drone along a forward-backward trajectory followed by a left-right trajectory. In experiment T3, we test the drone using random aggressive motions in Environment 2 and show the trajectory in Fig.5(b). Although there are features available in every direction, the RMSE of the gimbal camera shows the ability to perform stable pose estimation as compared to the static camera.

	Env	Len. (m)	RMSE	Mean	Std
Gim. T1	1	460.62	0.292	0.261	0.129
Gim. T2	1	180.59	0.310	0.272	0.132
Gim. T3	2	383.83	0.180	0.165	0.071
Stat. T1	1	460.62	0.384	0.347	0.164
Stat. T2	1	180.59	0.360	0.295	0.170
Stat. T3	2	383.82	0.414	0.383	0.157

Table 2: Mean and standard deviation of dynamic versus static camera VINS over two environments and three runs.

5.4. Online Dynamic Camera VINS calibration

To demonstrate online DC-VINS calibration, we test our method in Environment 2 and run our online approach over a trajectory of 456.19m. We initialize the calibration parameters within $\pm$3cm in translation, $\pm 10^\circ$ in rotation and add zero mean gaussian noise with a standard deviation of 7° to the true joint angle values. We compare our approach against the offline calibrated values as described in Sec.5.2 and only optimize for the joint angle values while keeping the offline parameters fixed. Error statistics of the two calibrated methods are reported in Table.3. The high RMSE error in the offline calibration method can be attributed to the poorly estimated calibration parameters remaining static emphasizing the need to gather new measurements and re-calibrate online. After calibration our method achieves an average translation error with mean 1.52cm and standard deviation 0.75cm and a rotation error with mean 1.11° and standard deviation 0.74° on the calibration parameters. We note that the sensitivity of the calibration parameters is largely dependent on the particular motion of the drone which helps in constraining certain parameters more than others and is the subject of future research. We argue that while the calibrated values do not reach the true values of the calibration chain, the ability to perform online calibration of a dynamic camera to an IMU is a significant step towards performing tightly-coupled visual-inertial odomtery and active vision in the field.

Calibration Method	RMSE	Mean	SD
Online (DC VINS)	0.7399	0.610	0.4186
Offline (DCC + Kalibr)	1.4056	1.274	0.592

Table 3: Error in online vs offline calibration.

6. Conclusion

In this paper, we demonstrate the ability to perform tightly coupled visual-inertial odometry with a dynamic camera while being able to recover the calibration parameters online and in flight. In order to avoid the degeneracy of certain calibration parameters as compared to previous approaches, we propose a minimal parameterization of the actuated chain between a dynamic camera and an IMU. We test our method in simulation and demonstrate the utility of a dynamic camera as compared to a static camera when performing VIO on an aerial vehicle undergoing aggressive motions. We compare our method against an offline DC calibration approach and show that our method reduces RMSE over the entire trajectory. Future work will look into investigating particular drone motions that help better constrain the calibration parameters in order to achieve a higher calibration accuracy needed for active visual-inertial odometry and SLAM.

References

[1] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015.

[2] Michael Bloesch, Hannes Sommer, Tristan Laidlow, Michael Burri, Gabriel Nuetzi, Peter Fankhauser, Dario Bellicoso, Christian Gehring, Stefan Leutenegger, Marco Hutter, and Roland Siegwart. A primer on the differential calculus of 3d orientations. In *https://arxiv.org/pdf/1606.05285.pdf*, October 2016.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Transactions on Robotics (T-RO)*, 32(6), 2016.

[4] Andrea Censi, Antonio Franchi, Luca Marchionni, and Giuseppe Oriolo. Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Transactions on Robotics (T-RO)*, 29(2), 2013.

[5] Christopher L. Choi, Jason Rebello, Leonid Koppel, Pranav Ganti, Arun Das, and Steven L Waslander. Encoderless Gimbal Calibration of Dynamic Multi-Camera Clusters. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[6] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research (IJRR)*, 18(3), 1999.

[7] Arun Das. Informed data selection for dynamic multi-camera clusters. In *PhD thesis, University of Waterloo, Mechanical and MEchatronics Engineering*, May 2018.

[8] Arun Das and Steven L. Waslander. Calibration of a dynamic camera cluster for multi-camera visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, October 2016.

[9] Guanglong Du and Ping Zhang. Imu-based online kinematic calibration of robot manipulator. In *The Scientific World Journal*, 2013.

[10] Kevin Eckenhoff, Patrick Geneva, Jesse Bloecker, , and Guoquan Huang. Multi-Camera Visual-Inertial Navigation with Online Intrinsic and Extrinsic Calibration. In *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, 2019.

[11] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.

[12] Fadri Furrer, Michael Burri, Markus Achtelik, and Roland Siegwart. *Robot Operating System (ROS): The Complete Reference (Volume 1)*, chapter RotorS—A Modular Gazebo MAV Simulator Framework, pages 595–625. Springer International Publishing, Cham, 2016.

[13] Michael Grupp. EVO: Python package for the evaluation of odometry and SLAM. `https://github.com/MichaelGrupp/evo`, 2017.

[14] Richard Scheunemann Hartenberg and Jacques Denavit. *Kinematic synthesis of linkages*. McGraw-Hill, 1964.

[15] R Hermann and A Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control (TACON)*, 22(5), 1977.

[16] Zheng Huai and Guoquan Huang. Robocentric visual-inertial odometry, 2018.

[17] Ganesh Iyer, R. KarnikRam, Krishna Murthy Jatavallabhula, and K. Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[18] Jonathan Kelly and Gaurav S Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.

[19] Jae-Hean Kim, Myung Jin Chung, and Byung Tae Choi. Recursive estimation of motion and a scene model with a two-camera system of divergent view. *Pattern Recognition*, 43:2265–2280, 2010.

[20] J S Kim and T Kanade. Degeneracy of the linear seventeen-point algorithm for generalized essential matrix. *Journal of Mathematical Imaging and Vision*, 37(1):40–48, May 2010.

[21] Sin-Jung Kim, Mun-Ho Jeong, Joong-Jae Lee, Ji-Yong Lee, KangGeon Kim, Bum-Jae You, and Sang-Rok Oh. Robot head-eye calibration using the minimum variance method. In *2010 IEEE International Conference on Robotics and Biomimetics, ROBIO 2010, Tianjin, China, December 14-18, 2010*, pages 1446–1451, 2010.

[22] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Timothy Furgale. Keyframe-based Visual-Inertial odometry using nonlinear optimization. *The International Journal of Robotics Research (IJRR)*, 34(3), 2015.

[23] Jinghui Li, Akitoshi Ito, Hiroyuki Yaguchi, and Yusuke Maeda. Simultaneous kinematic calibration, localization, and mapping (SKCLAM) for industrial robot manipulators. volume 33, 2019.

[24] Anastasios I. Mourikis and Stergios I. Roumeliotis. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, New York, NY, April 2007.

[25] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. 2017.

[26] Bhavit Patel, Michael Warren, and Angela Schoellig. Point Me In The Right Direction: Improving Visual Localization on UAVs with Active Gimballed Camera Pointing. In *Computer and Robot Vision (CRV)*, Kingston, Canada, 2019.

[27] Vijay Pradeep, Kurt Konolige, and Eric Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *International Symposium on Experimental Robotics (ISER)*, 12/2010 2010.

[28] Tong Qin, Shaozu Cao, Jie Pan, and Shaojie Shen. A general optimization-based framework for global pose estimation with multiple sensors, 2019.

[29] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics (T-RO)*, 34(4), 2018.

[30] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A general optimization-based framework for local odometry estimation with multiple sensors, 2019.

[31] Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE, 2018.

[32] Jason Rebello, Arun Das, and Steven L Waslander. Autonomous active calibration of a Dynamic Camera Cluster using Next-Best-View. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[33] Jason Rebello, Angus Fung, and Steven L Waslander. AC/DCC : Accurate Calibration of Dynamic Camera Clusters for Visual SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[34] Joern Rehder and Roland Siegwart. Camera/IMU Calibration Revisited. *IEEE Sensors Journal*, 17(11), 2017.

[35] N. Schneider, Florian Piewak, C. Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[36] Jieying Shi, Ziheng Zhu, Jianhua Zhang, Ruyu Liu, Zhenhua Wang, Shengyong Chen, and Honghai Liu. CalibRCNN: Calibrating Camera and LiDAR by Recurrent Convolutional Neural Network and Geometric Constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, 2020.

[37] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. Openvslam: A versatile visual slam framework. 2019.

[38] Michael J. Tribou, David W. Wang, and Steven L. Waslander. Degenerate motions in multicamera cluster slam with non-overlapping fields of view. *Image and Vision Computing*, 50:27–41, 2016.

[39] Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Direct visual-inertial odometry with stereo cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016.

[40] Michael Warren, Angela Schoellig, and Timothy D. Barfoot. Level-Headed: Evaluating Gimbal-Stabilised Visual Teach and Repeat for Improved Localisation Performance. In *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018.

[41] E. Wise, M. Giamou, S. Khoubyarian, A. Grover, and J. Kelly. Certifiably optimal monocular hand-eye calibration. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, virtual, September 2020.

[42] Yulin Yang, Patrick Geneva, Kevin Eckenhoff, and Guoquan Huang. Degenerate Motion Analysis for Aided INS with Online Spatial and Temporal Calibration. *IEEE Robotics and Automation Letters (RA-L)*, 4(2), 2019.

[43] Ganning Zhao, Jiesi Hu, Suya You, and C.-C. Jay Kuo. CalibDNN: multimodal sensor calibration for perception using deep neural networks. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXX*, volume 11756, 2021.