Fusing Inertial Sensor Data in an Extended Kalman Filter for 3D Camera Tracking

A. Tanju Erdem, *Member, IEEE*, and Ali O. Ercan, *Member, IEEE*

***Abstract*—In a setup where camera measurements are used to estimate 3D ego-motion in an Extended Kalman Filter (EKF) framework, it is well known that inertial sensors (i.e., accelerometers and gyroscopes) are especially useful when the camera undergoes fast motion. Inertial sensor data can be fused at the EKF with the camera measurements in either the correction stage (as measurement inputs) or the prediction stage (as control inputs). In general, only one type of inertial sensor is employed in the EKF in the literature, or when both are employed they are both fused in the same stage. In this paper, we provide an extensive performance comparison of every possible combination of fusing accelerometer and gyroscope data as control or measurement inputs using the same data set collected at different motion speeds. In particular, we compare the performances of different approaches based on 3D pose errors, in addition to camera reprojection errors commonly found in the literature, which provides further insight into the strengths and weaknesses of different approaches. We show using both simulated and real data that it is always better to fuse both sensors in the measurement stage and that in particular, accelerometer helps more with the 3D position tracking accuracy whereas gyroscope helps more with the 3D orientation tracking accuracy. We also propose a simulated data generation method, which is beneficial for the design and validation of tracking algorithms involving both camera and IMU measurements in general.**

***Index Terms*—Inertial sensor fusion, Extended Kalman Filter, 3D camera tracking, inertial measurement unit, accelerometer, gyroscope.**

I. Introduction

ACCURATE 3D tracking is important for many applications including navigation, visualization, human-computer interaction and augmented reality [1]. Although there are various methods proposed for 3D tracking, those that use GPS or cellular technologies are not suitable for indoor applications [2]. Methods using IR light and RF signals require the placement of IR light emitters or RFID tags on the scene [3], which may not be acceptable, or even possible, for some applications such as cultural heritage. Computer vision based tracking methods that rely on camera measurements only, do not possess these problems and perform well only at slow motion [4]. However, fast camera motion may result in blurred features that may not be localized accurately, thereby resulting in degradation in estimated tracking accuracy. Inertial sensors (*i.e.*, accelerometers and gyroscopes), on the other hand, measure the derivatives of motion and their signals are more reliable at fast motion since their SNR improves with the amount of motion. However, 3D pose estimation using inertial sensors alone suffers from drift [5]. Thus, it is suggested to fuse inertial sensor data with camera measurements for 3D tracking [6][7].

There are many approaches to fuse the inertial sensor data with camera data in the literature [8][9][10]. One popular approach is to fuse them in an Extended Kalman Filter (EKF) [10][11][12][13]. EKF has two stages, namely the time update (*i.e.*, prediction) stage and the measurement update (*i.e.*, correction) stage. Hence, there are two alternative ways to fuse inertial sensor data in an EKF: one option is to use inertial sensor data at the correction stage, which we refer to as using inertial sensor data as *measurement* input, and the second option, is to use inertial sensor data at the prediction stage, which we refer to as using inertial sensor data as *control* input. Therefore, there are a total of eight possible approaches for fusing accelerometer and gyroscope data in an EKF framework: both used as control inputs, both used as measurement inputs, one is used as control input while the other one is used as measurement input, and finally, only one is used as control or measurement input while the other one is not used.

Five of the above eight combinations, namely, fusing both inertial sensor data as measurement or control inputs, fusing only accelerometer as measurement or control input, and fusing only gyroscope as measurement have been investigated in the literature [12][13][14]. Ref. [12] compares three of these cases, namely, both inertial sensor data fused as measurement inputs, both fused as control inputs, and only gyroscope data fused as measurement input, and suggests that all three cases perform similarly well at fast and slow speeds except that gyroscope only as measurement input case results in poor tracking quality at fast speeds. Ref. [13] compares two cases, namely, both inertial sensor data fused as measurement inputs and only gyroscope data fused as measurement input, and concludes that the case of fusing both sensor data significantly outperforms the gyroscope only case. Our previous work [14] fuses only accelerometer data, and suggests that fusing accelerometer data either as measurement or as control input brings about similar improvement to tracking accuracy. To the best of our knowledge, the three remaining cases, *i.e.*, only gyroscope data fused as control input, gyroscope data fused as

Manuscript is submitted for review on 5 August 2013. This work was supported in part by TÜBİTAK under grant EEEAG-110E053.

A. T. Erdem and A. O. Ercan are with Ozyegin University, Istanbul, Turkey (e-mails: {tanju.erdem, ali.ercan}@ozyegin.edu.tr).

measurement input while accelerometer data fused as control inputs, and accelerometer data fused as measurement input while gyroscope data fused as control input are not covered in the literature before. Furthermore, there is no previous work that compares all eight cases using the same data set.

In this paper, we compare, using both realistic extensive simulations and real data, the tracking accuracies of all possible eight configurations of fusing inertial sensor data at different motion speeds. In addition to the common approach of using 2D reprojection errors, we use 3D pose errors to compare the performances of these configurations, which provides further insight into the strengths and weaknesses of different tracking approaches.

In Section 2, we present background information regarding EKF, use of quaternions, and camera-inertial sensor set-up. In Section 3, we provide for the first time in the literature a complete set of EKF equations including Jacobian matrices corresponding to all cases of camera-inertial sensor fusion approaches. In Section 4, we describe our simulation setup. In Sections 5 and 6 we present the results of our simulations and real experiments, respectively. Discussions and conclusions are given in Section 7.

II. Preliminaries

A. Extended Kalman Filter Equations

In the following, we provide the EKF equations for the purpose of completeness and establishing the notation. We make discrete-time assumption, since measurements are obtained in discrete intervals. Let

$$X_t = f(X_{t-1}, u_t, \rho_t, \zeta_t) \quad \text{and} \quad y_t = h(X_t, \eta_t) \tag{1}$$

represent the process and measurement models, respectively, where X_t denotes the state vector, u_t denotes the control input, and y_t denotes the measurement vector, all at time t. ρ_t, ζ_t, and η_t denote the process, control, and measurement noise vectors, which are assumed to be zero-mean white Gaussian noise processes independent of each other with sample covariance matrices Γ, Π, and Σ, respectively. When there is no control input, the terms u and ζ are dropped from (1).

Depending on which process model is used, the content of the state X changes. For example, when accelerometer is used as measurements, the state has to include the acceleration since the measurement model requires it. On the other hand, if it is used as control input, we do not include acceleration in the state for the sake of reducing the computational complexity [14]. Similarly, the angular velocity is included (*vs*. not included) in the state when gyroscope data are used as measurements (*vs*. control inputs). Define

$$\begin{aligned} \hat{x}_t &= \mathrm{E}(X_t | y_1, y_2, \dots, y_{t-1}), \\ x_t &= \mathrm{E}(X_t | y_1, y_2, \dots, y_t), \\ \hat{P}_t &= \mathrm{E}(X_t X_t^T | y_1, y_2, \dots, y_{t-1}) - \hat{x}_t \hat{x}_t^T, \\ P_t &= \mathrm{E}(X_t X_t^T | y_1, y_2, \dots, y_t) - x_t x_t^T, \end{aligned} \tag{2}$$

where $\mathrm{E}(\cdot)$ denotes the expectation operator. Then, starting with initial state mean x_0 and covariance matrix P_0, the time update equations for state mean x and state covariance matrix P are given as:

$$\begin{aligned} \hat{x}_t &= f(x_{t-1}, u_t, \rho_t, \zeta_t)|_{\rho_t, \zeta_t = 0}, \\ \hat{P}_t &= F_t P_{t-1} F_t^T + V_t \Gamma_t V_t^T + L_t \Pi_t L_t^T, \end{aligned} \tag{3}$$

where the Jacobians are defined as

$$\begin{aligned} F_t &= \frac{\partial f}{\partial X}|_{X = x_{t-1}, u = u_t, \rho, \zeta = 0}, \quad V_t = \frac{\partial f}{\partial \rho}|_{X = x_{t-1}, u = u_t, \rho, \zeta = 0}, \\ &\text{and} \quad L_t = \frac{\partial f}{\partial \zeta}|_{X = x_{t-1}, u = u_t, \rho, \zeta = 0}. \end{aligned} \tag{4}$$

When there is no control input, the variables u_t, ζ_t are dropped from (3) and (4), and consequently, L_t vanishes.

The Kalman gain K_t and innovation z_t are calculated as

$$\begin{aligned} K_t &= \hat{P}_t H_t^T S_t^{-1}, \quad \text{where} \quad S_t = H_t \hat{P}_t H_t^T + \Sigma_t, \\ z_t &= y_t - h(\hat{x}_t, \eta_t)|_{\eta_t = 0}, \end{aligned} \tag{5}$$

and the measurement update equations for state mean and state covariance matrix are given as

$$x_t = \hat{x}_t + K_t z_t \quad \text{and} \quad P_t = \hat{P}_t - K_t H_t \hat{P}_t, \tag{6}$$

where the measurement Jacobian is defined as

$$H_t = \frac{\partial h}{\partial X}|_{X = \hat{x}_{t-1}, \eta = 0} \tag{7}$$

In the rest of the paper, we will omit the time index for brevity whenever it is clear from the context.

B. Representation of Rotation in EKF

One of the components of the state vector X is a unit quaternion representing the orientation of the camera. A unit quaternion q is a 4D vector composed of a number λ and a 3D vector φ defined as

$$q = [\lambda \quad \varphi^T]^T, \quad \text{where} \quad \lambda^2 + \varphi \cdot \varphi = 1. \tag{8}$$

A unit quaternion represents a rotation by an angle θ around a unit axis $\hat{n}$, which are related to the quaternion components as

$$\lambda = cos\frac{\theta}{2} \quad \text{and} \quad \varphi = sin\frac{\theta}{2}\hat{n}. \tag{9}$$

The corresponding rotation matrix R is obtained using the formula [15]

$$R = I + 2\lambda[\varphi]_\times + 2[\varphi]_\times^2 \tag{10}$$

where $[\varphi]_\times$ represents a matrix that implements a cross product with φ, i.e., $[\varphi]_\times \sigma = \varphi \times \sigma$.

It is also possible to represent a rotation in terms of angles of rotation $\delta_{\theta,x}$, $\delta_{\theta,y}$, and $\delta_{\theta,z}$, around coordinate axes x, y, and z, respectively. Resulting orientation depends on the order of rotations around the coordinate axes, however, this dependence nearly disappears if the angles are small, and the corresponding unit quaternion is approximately given as

$$\delta_q \approx \left[cos\frac{\|\delta_\theta\|}{2} \quad \frac{\delta_\theta^T}{\|\delta_\theta\|} sin\frac{\|\delta_\theta\|}{2} \right]^T, \tag{11}$$

where $\delta_\theta = [\delta_{\theta,x} \quad \delta_{\theta,y} \quad \delta_{\theta,z}]^T$. If the angles are very small, the above approximation can be further simplified to

$$\delta_q \approx \left[1 \quad \frac{1}{2}\delta_\theta^T \right]^T. \tag{12}$$

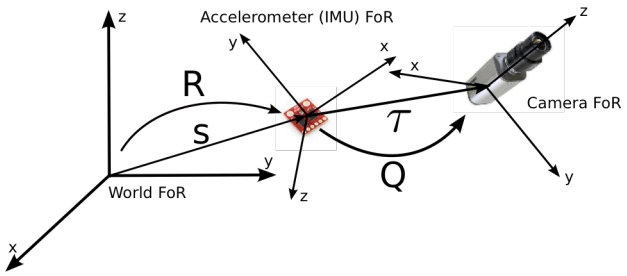

Figure 1. World, IMU, and camera coordinate systems and their relative positions and orientations.

Although this further simplification results in a non-unit quaternion, it proves to be very useful in simplifying EKF calculations. Note that, after the prediction and correction stages of the EKF, the quaternion in the state vector has to be normalized, and the state covariance has to be updated accordingly as in [12].

C. Camera-IMU Setup and Camera Measurement Model

We use a setup where a camera and an inertial measurement unit (IMU) containing an accelerometer and a gyroscope that are rigidly fixed together. Since gyroscope measures angular velocity, its readings are independent of its location. Therefore, the relative location of the gyroscope with respect to the accelerometer and/or the camera does not need to be known or taken into account in EKF equations. Without loss of generality, we assume that the accelerometer and the gyroscope have the same orientation, therefore, we call their poses simultaneously as the IMU pose.

EKF equations turn out to be simpler if one uses the IMU pose in the state vector rather than that of the camera. Let s denote the position of the IMU in World FoR and R represent the rotation from the world frame of reference (FoR) to IMU FoR (see Figure 1). Note that R is the rotation matrix corresponding to quaternion in state X. Also let Q represent the rotation from IMU FoR to camera FoR and τ denote the IMU to camera displacement in IMU FoR. We assume Q and τ are obtained in a calibration step and the tracking problem becomes estimating R and s over time.

We assume that a 3D map, *i.e.* 3D coordinates κ of a set of feature points, of the scene is available. Such a map can be obtained from a recorded video of the scene prior to tracking [14]. During tracking, these feature points are detected in the captured images and their 2D positions on the image plane become the camera measurements μ_t. In the EKF, we assume a pinhole camera model for these measurements:

$$\mu_t = \begin{bmatrix} fp_{1,t}/p_{3,t} \\ fp_{2,t}/p_{3,t} \end{bmatrix} + \varepsilon_{\mu,t}, \tag{13}$$

where f is the focal length of the camera, and $\varepsilon_{\mu,t}$ denotes the camera measurement noise, all in pixels, and p_t is the position of a 3D scene point κ in camera FoR:

$$p_t = [p_{1,t} \quad p_{2,t} \quad p_{3,t}]^T = Q(R_t(\kappa - s_t) - \tau). \tag{14}$$

Such calibrated 2D measurements can be obtained after correcting images for any possible lens distortions, optical center shifts, and non-rectangular and skewed pixels [15].

III. IMU Fusion Approaches for EKF

Inertial sensor data can be included in an EKF as control or measurement inputs. We have implemented EKFs for all possible combinations of using gyroscopes and accelerometers as control or measurement inputs. Note that camera and inertial sensors may have different sampling rates. While EKF performs both prediction and correction when a measurement input arrives, it performs only correction when a control input arrives.

As a shorthand notation, we use a three-letter abbreviation. As there is no direct way of fusing camera measurements as control inputs in the EKF, they are always used as measurements, therefore the first letter in the three-letter abbreviation is always "M". The second and third letters indicate whether accelerometer and gyroscope data, respectively, are used as control input (C), or as measurement (M), or not used (X).

In the following, we give the detailed expressions for process and measurement equations, as well as the expressions for Jacobian matrices, for all possible fusion approaches. The derivation steps of these matrices are omitted due to space considerations. Sensor measurements are represented in their own frames of reference.

A. Camera Only Approach (MXX)

This case is provided to serve as a baseline for the various fusion approaches. In the camera-only approach, neither accelerometer nor gyroscope data are employed. Therefore, there are only camera measurements and no control inputs. In this case, EKF process and measurement variables become

$$X = \begin{bmatrix} s \\ v \\ q \end{bmatrix}, \quad y = [\mu], \quad \rho = \begin{bmatrix} \varepsilon_v \\ \varepsilon_\theta \end{bmatrix}, \quad \eta = [\varepsilon_\mu], \tag{15}$$

where s and v stand for the state variables corresponding to the 3D position and velocity of the IMU in the World FoR, and q represents the orientation quaternion corresponding to rotation matrix R, μ stands for 2D camera measurements, ε_v and ε_θ stand for velocity and angle process noises, and ε_μ stands for camera measurement noise. The process and measurement noises are all zero mean white Gaussian noises independent of each other (determination of their variances is explained in Sections IV and V).

The process equations in (1) are given as follows:

$$\begin{aligned} s_t &= s_{t-1} + Tv_{t-1} + T\varepsilon_{v,t}, \\ v_t &= v_{t-1} + \varepsilon_{v,t}, \\ q_t &= \delta_{q,t} \odot q_{t-1}, \end{aligned} \tag{16}$$

where $\odot$ denotes quaternion product and $\delta_{q,t}$ is the time change in orientation quaternion. Using the small angle approximation (12), the above process equation for the quaternion can be expressed in terms of the scalar and vector components of the quaternion as

$$\begin{aligned} \lambda_t &= \lambda_{t-1} - \tfrac{1}{2}\varphi_{t-1}^T \delta_{\theta,t}, \\ \varphi_t &= \varphi_{t-1} + \tfrac{1}{2}\lambda_{t-1}\delta_{\theta,t} - \tfrac{1}{2}\varphi_{t-1} \times \delta_{\theta,t}, \end{aligned} \tag{17}$$

where we used the quaternion product formula [15]. In the following, we will simply give the expressions for $\delta_{\theta,t}$ in

order to specify the process equation for the quaternion for different camera-IMU fusion approaches and implicitly assume the prediction model in (17) for all cases but with different δ_θ. Thus, for the camera only case we have

$$\delta_{\theta,t} = \varepsilon_{\theta,t}. \tag{18}$$

The only measurement equation in this case is that of camera measurements (14).

State Jacobian matrix F for this case is given by

$$F = \begin{bmatrix} F_1 & 0_{6\times4} \\ 0_{4\times6} & I_{4\times4} \end{bmatrix}, \text{ where } F_1 = \begin{bmatrix} I_{3\times3} & TI_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}. \tag{19}$$

Process noise Jacobian matrix V is calculated as

$$V = \begin{bmatrix} V_1 & 0_{6\times3} \\ 0_{4\times3} & V_2 \end{bmatrix}, \text{ where}$$
$$V_1 = \begin{bmatrix} TI_{3\times3} \\ I_{3\times3} \end{bmatrix} \text{ and } V_2 = \begin{bmatrix} -\frac{1}{2}\varphi^T \\ \frac{1}{2}\lambda I - \frac{1}{2}[\varphi]_\times \end{bmatrix}. \tag{20}$$

This case does not involve a control noise Jacobian matrix L as it does not employ a control input. Measurement Jacobian matrix H is given as follows:

$$H = H_1 Q[-R \quad 0_{3\times3} \quad H_2(\kappa - s)], \tag{21}$$

where

$$H_1 = f\begin{bmatrix} \frac{1}{p_3} & 0 & -\frac{p_1}{p_3^2} \\ 0 & \frac{1}{p_3} & -\frac{p_2}{p_3^2} \end{bmatrix}, \tag{22}$$

and

$$H_2(\sigma) = [2[\varphi]_\times\sigma \quad -2(\lambda[\sigma]_\times + [[\varphi]_\times\sigma]_\times + [\varphi]_\times[\sigma]_\times)]. \tag{23}$$

B. Accelerometer As Control (MCX) [14]

In this approach, accelerometer data γ are used as control inputs, gyroscope data are not employed, and only camera data are used as measurements. In this case, EKF process and measurement variables become:

$$X = \begin{bmatrix} s \\ v \\ q \end{bmatrix}, \quad y = [\mu], u = [\gamma], \quad \rho = \begin{bmatrix} \varepsilon_a \\ \varepsilon_\theta \end{bmatrix}, \quad \zeta = [\varepsilon_\gamma], \quad \eta = [\varepsilon_\mu]. \tag{24}$$

The process equations are given as follows, where position and velocity terms include accelerometer data as control inputs:

$$\begin{gathered} s_t = s_{t-1} + Tv_{t-1} + \frac{1}{2}T^2\left(R_{t-1}^T(\gamma + \varepsilon_\gamma) - g\right) + \frac{1}{2}T^2\varepsilon_{a,t}, \\ v_t = v_{t-1} + T\left(R_{t-1}^T(\gamma + \varepsilon_\gamma) - g\right) + T\varepsilon_{a,t}, \\ \delta_{\theta,t} = \varepsilon_{\theta,t}, \end{gathered} \tag{25}$$

where g is the gravity in World FoR. The only measurement equation is that of camera measurements (15).

Jacobian matrices for this case are calculated as follows:

$$F = \begin{bmatrix} F_1 & F_2 \\ 0_{4\times6} & I_{4\times4} \end{bmatrix}, \tag{26}$$

where

$$F_2 = \begin{bmatrix} T^2[\varphi]_\times\gamma & T^2(\lambda[\gamma]_\times - [\varphi]_\times[\gamma]_\times - [[\varphi]_\times\gamma]_\times) \\ 2T[\varphi]_\times\gamma & 2T(\lambda[\gamma]_\times - [\varphi]_\times[\gamma]_\times - [[\varphi]_\times\gamma]_\times) \end{bmatrix}. \tag{27}$$

Process noise Jacobian matrix V is given by

$$V = \begin{bmatrix} V_3 & 0_{6\times3} \\ 0_{4\times3} & V_2 \end{bmatrix}, \quad \text{where} \quad V_3 = \begin{bmatrix} \frac{1}{2}T^2I_{3\times3} \\ TI_{3\times3} \end{bmatrix}. \tag{28}$$

while control noise Jacobian matrix L is calculated as

$$L = \begin{bmatrix} L_1 \\ 0_{4\times3} \end{bmatrix}, \quad \text{where} \quad L_1 = \begin{bmatrix} \frac{1}{2}T^2R^T \\ TR^T \end{bmatrix}. \tag{29}$$

Finally, measurement Jacobian matrix H is given by (21).

C. Accelerometer As Measurement (MMX) [14]

In this approach, accelerometer data are used as measurements, whereas gyroscope data are not employed. Since accelerometer data are used as measurement input, acceleration of the IMU in the World FoR is included in the state vector. EKF process and measurement variables are:

$$X = \begin{bmatrix} s \\ v \\ a \\ q \end{bmatrix}, \quad y = \begin{bmatrix} \mu \\ \gamma \end{bmatrix}, \quad \rho = \begin{bmatrix} \varepsilon_a \\ \varepsilon_\theta \end{bmatrix}, \quad \eta = \begin{bmatrix} \varepsilon_\mu \\ \varepsilon_\gamma \end{bmatrix}. \tag{30}$$

The process equations are given as follows:

$$\begin{gathered} s_t = s_{t-1} + Tv_{t-1} + \frac{1}{2}T^2a_{t-1} + \frac{1}{2}T^2\varepsilon_{a,t}, \\ v_t = v_{t-1} + Ta_{t-1} + T\varepsilon_{a,t}, \\ a_t = a_{t-1} + \varepsilon_{a,t}, \\ \delta_{\theta,t} = \varepsilon_{\theta,t}. \end{gathered} \tag{31}$$

In addition to camera measurement equation (13), there is an IMU measurement equation involving accelerometer data:

$$\gamma_t = R_t(a_t + g) + \varepsilon_{\gamma,t}. \tag{32}$$

Jacobian matrices for this case are calculated as follows:

$$F = \begin{bmatrix} F_3 & 0_{9\times4} \\ 0_{4\times9} & I_{4\times4} \end{bmatrix}, \tag{33}$$

where

$$F_3 = \begin{bmatrix} I_{3\times3} & TI_{3\times3} & \frac{1}{2}T^2I_{3\times3} \\ 0_{3\times3} & I_{3\times3} & TI_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}. \tag{34}$$

Process noise Jacobian matrix V is calculated as

$$V = \begin{bmatrix} V_4 & 0_{9\times3} \\ 0_{4\times3} & V_2 \end{bmatrix}, \quad \text{where} \quad V_4 = \begin{bmatrix} \frac{1}{2}T^2I_{3\times3} \\ TI_{3\times3} \\ I_{3\times3} \end{bmatrix}. \tag{35}$$

This case does not involve control noise Jacobian matrix L as it does not employ a control input. Measurement Jacobian matrix H is given by

$$\begin{gathered} H = \begin{bmatrix} H_3 \\ H_4 \end{bmatrix}, \text{ where } H_3 = H_1 Q[-R \quad 0_{3\times6} \quad H_2(\kappa - s)], \\ \text{and} \quad H_4 = [0_{3\times6} \quad R \quad H_2(a + g)] \end{gathered} \tag{36}$$

D. Gyroscope As Control (MXC)

In this approach, gyroscope data β are used as control inputs and accelerometer data are not employed. EKF process and measurement variables are:

$$X = \begin{bmatrix} s \\ v \\ q \end{bmatrix},\ y = [\mu],\ u = [\beta],\ \rho = \begin{bmatrix} \varepsilon_v \\ \varepsilon_\omega \end{bmatrix},\ \zeta = [\varepsilon_\beta],\ \eta = [\varepsilon_\mu]. \quad (37)$$

The process equations are given as follows, where quaternion update involves gyroscope data as control inputs:

$$\begin{aligned} s_t &= s_{t-1} + Tv_{t-1} + T\varepsilon_{v,t}, \\ v_t &= v_{t-1} + \varepsilon_{v,t}, \\ \delta_{\theta,t} &= T(\beta_t + \varepsilon_{\beta,t}) + T\varepsilon_{\omega,t}. \end{aligned} \quad (38)$$

The only measurement equation is that of camera measurements.

Jacobian matrices for this case are calculated as follows:

$$F = \begin{bmatrix} F_1 & 0_{6\times4} \\ 0_{4\times6} & F_4 \end{bmatrix}, \quad (39)$$

where

$$F_4 = \begin{bmatrix} 1 & -\frac{1}{2}T\beta^T \\ \frac{1}{2}T\beta & I + \frac{1}{2}T[\beta]_\times \end{bmatrix} \quad (40)$$

Process noise Jacobian matrix V is calculated as

$$V = \begin{bmatrix} V_1 & 0_{6\times3} \\ 0_{4\times3} & TV_2 \end{bmatrix}. \quad (41)$$

Control noise Jacobian matrix L is calculated as

$$L = \begin{bmatrix} 0_{6\times3} \\ TV_2 \end{bmatrix} \quad (42)$$

Finally, measurement Jacobian matrix H is given as

$$H = H_1 Q[-R \quad 0_{3\times3} \quad H_2(\kappa - s)]. \quad (43)$$

E. Gyroscope As Measurement (MXM) [12, 13]

In this approach, in addition to camera data, gyroscope data are also used as measurements, whereas accelerometer data are not employed. Since gyroscope data are used as measurement input, angular velocity of the IMU in the IMU FoR is included in the state vector. EKF process and measurement variables are:

$$X = \begin{bmatrix} s \\ v \\ q \\ \omega \end{bmatrix}, \quad y = \begin{bmatrix} \mu \\ \beta \end{bmatrix}, \quad \rho = \begin{bmatrix} \varepsilon_v \\ \varepsilon_\omega \end{bmatrix}, \quad \eta = \begin{bmatrix} \varepsilon_\mu \\ \varepsilon_\beta \end{bmatrix}. \quad (44)$$

The process equations are given as follows:

$$\begin{aligned} s_t &= s_{t-1} + Tv_{t-1} + T\varepsilon_{v,t}, \\ v_t &= v_{t-1} + \varepsilon_{v,t}, \\ \delta_{\theta,t} &= T\omega_{t-1} + T\varepsilon_{\omega,t}, \\ \omega_t &= \omega_{t-1} + \varepsilon_{\omega,t}. \end{aligned} \quad (45)$$

In addition to camera measurement equation, there is an IMU measurement equation involving gyroscope data:

$$\beta_t = \omega_t + \varepsilon_{\beta,t}. \quad (46)$$

Jacobian matrices for this case are calculated as follows:

$$F = \begin{bmatrix} F_1 & 0_{6\times7} \\ 0_{7\times6} & F_5 \end{bmatrix}, \quad (47)$$

where

$$F_5 = \begin{bmatrix} 1 & -\frac{1}{2}T\omega^T & -\frac{1}{2}T\varphi^T \\ \frac{1}{2}T\omega & I + \frac{1}{2}T[\omega]_\times & \frac{1}{2}T(\lambda I - [\varphi]_\times) \\ 0_{3\times1} & 0_{3\times3} & I_{3\times3} \end{bmatrix}. \quad (48)$$

Process noise Jacobian matrix V is calculated as

$$V = \begin{bmatrix} V_1 & 0_{6\times3} \\ 0_{7\times3} & V_2' \end{bmatrix}, \quad \text{where} \quad V_2' = \begin{bmatrix} TV_2 \\ I_{3\times3} \end{bmatrix}. \quad (49)$$

This case does not involve a control noise Jacobian matrix L as it does not employ a control input. Measurement Jacobian matrix H is calculated as

$$\begin{aligned} H &= \begin{bmatrix} H_5 \\ H_6 \end{bmatrix}, \quad \text{where} \\ H_5 &= H_1 Q[-R \quad 0_{3\times3} \quad H_2(\kappa - s) \quad 0_{3\times3}], \quad \text{and} \\ H_6 &= [0_{3\times10} \quad I]. \end{aligned} \quad (50)$$

F. Accelerometer and Gyroscope As Control (MCC) [12]

In this approach, both accelerometer and gyroscope data are used as control inputs, hence, neither acceleration nor angular velocity appear in the state vector. EKF process and measurement variables are:

$$X = \begin{bmatrix} s \\ v \\ q \end{bmatrix},\ y = [\mu],\ u = \begin{bmatrix} \gamma \\ \beta \end{bmatrix},\ \rho = \begin{bmatrix} \varepsilon_a \\ \varepsilon_\omega \end{bmatrix},\ \zeta = \begin{bmatrix} \varepsilon_\gamma \\ \varepsilon_\beta \end{bmatrix},\ \eta = [\varepsilon_\mu]. \quad (51)$$

The process equations are given as follows where position and velocity update equations involve accelerometer data while quaternion update involves gyroscope data:

$$\begin{aligned} s_t &= s_{t-1} + Tv_{t-1} + \frac{1}{2}T^2(R_{t-1}^T(\gamma + \varepsilon_\gamma) - g) + \frac{1}{2}T^2\varepsilon_{a,t}, \\ v_t &= v_{t-1} + T(R_{t-1}^T(\gamma + \varepsilon_\gamma) - g) + T\varepsilon_{a,t}, \\ \delta_{\theta,t} &= T(\beta_t + \varepsilon_{\beta,t}) + T\varepsilon_{\omega,t}. \end{aligned} \quad (52)$$

The only measurement equation is that of camera measurements.

Jacobian matrices for this case are given as follows:

$$F = \begin{bmatrix} F_1 & F_2 \\ 0_{4\times6} & F_4 \end{bmatrix},\ V = \begin{bmatrix} V_3 & 0_{6\times3} \\ 0_{4\times3} & TV_2 \end{bmatrix},\ L = \begin{bmatrix} L_1 & 0_{6\times3} \\ 0_{4\times3} & TV_2 \end{bmatrix}, \quad (53)$$

and,

$$H = H_1 Q[-R \quad 0_{3\times3} \quad H_2(\kappa - s)]. \quad (54)$$

G. Accelerometer As Control, Gyroscope As Measurement (MCM)

In this approach, accelerometer data are used as control inputs and gyroscope data are used as measurements. Hence, although angular velocity appears in the state vector, acceleration does not. EKF process and measurement variables are:

$$X = \begin{bmatrix} s \\ v \\ q \\ \omega \end{bmatrix},\ y = \begin{bmatrix} \mu \\ \beta \end{bmatrix}, u = [\gamma],\ \rho = \begin{bmatrix} \varepsilon_a \\ \varepsilon_\omega \end{bmatrix},\ \zeta = [\varepsilon_\gamma],\ \eta = \begin{bmatrix} \varepsilon_\mu \\ \varepsilon_\beta \end{bmatrix}. \quad (55)$$

The process equations are given as follows where position and velocity update involves accelerometer data:

$$\begin{aligned} s_t &= s_{t-1} + Tv_{t-1} + \tfrac{1}{2}T^2\left(R_{t-1}^T(\gamma + \varepsilon_\gamma) - g\right) + \tfrac{1}{2}T^2\varepsilon_{a,t}, \\ v_t &= v_{t-1} + T\left(R_{t-1}^T(\gamma + \varepsilon_\gamma) - g\right) + T\varepsilon_{a,t}, \\ \delta_{\theta,t} &= T\omega_{t-1} + T\varepsilon_{\omega,t}, \\ \omega_t &= \omega_{t-1} + \varepsilon_{\omega,t}. \end{aligned} \quad (56)$$

In addition to camera measurement equation, there is an IMU measurement equation involving gyroscope data:

$$\beta_t = \omega_t + \varepsilon_{\beta,t}. \quad (57)$$

Jacobian matrices for this case are calculated as follows:

$$F = \begin{bmatrix} F_1 & F_2' \\ 0_{7\times 6} & F_5 \end{bmatrix}, \quad \text{where} \quad F_2' = [F_2 \quad 0_{6\times 3}], \quad (58)$$

and

$$V = \begin{bmatrix} V_3 & 0_{6\times 3} \\ 0_{7\times 3} & V_2' \end{bmatrix},\ L = \begin{bmatrix} L_1 \\ 0_{7\times 3} \end{bmatrix},\ \text{and}\ H = \begin{bmatrix} H_5 \\ H_6 \end{bmatrix}. \quad (59)$$

H. Accelerometer As Measurement, Gyroscope As Control (MMC)

In this approach, accelerometer data are used as measurements and gyroscope data are used as control inputs. Hence, acceleration appears in the state vector, while angular velocity does not. EKF process and measurement variables are:

$$X = \begin{bmatrix} s \\ v \\ a \\ q \end{bmatrix},\ y = \begin{bmatrix} \mu \\ \gamma \end{bmatrix},\ u = [\beta],\ \rho = \begin{bmatrix} \varepsilon_a \\ \varepsilon_\omega \end{bmatrix},\ \zeta = [\varepsilon_\beta],\ \eta = \begin{bmatrix} \varepsilon_\mu \\ \varepsilon_\gamma \end{bmatrix}. \quad (60)$$

The process equations are given as follows, where quaternion update involves gyroscope data:

$$\begin{aligned} s_t &= s_{t-1} + Tv_{t-1} + \tfrac{1}{2}T^2a_{t-1} + \tfrac{1}{2}T^2\varepsilon_{a,t}, \\ v_t &= v_{t-1} + Ta_{t-1} + T\varepsilon_{a,t}, \\ a_t &= a_{t-1} + \varepsilon_{a,t}, \\ \delta_{\theta,t} &= T(\beta_t + \varepsilon_{\beta,t}) + T\varepsilon_{\omega,t}. \end{aligned} \quad (61)$$

In addition to camera measurement equation, there is an IMU measurement equation involving accelerometer data:

$$\gamma_t = R_t(a_t + g) + \varepsilon_{\gamma,t}. \quad (62)$$

Jacobian matrices for this case are calculated as follows:

$$F = \begin{bmatrix} F_3 & 0_{9\times 4} \\ 0_{4\times 9} & F_4 \end{bmatrix},\ V = \begin{bmatrix} V_4 & 0_{9\times 3} \\ 0_{4\times 3} & TV_2 \end{bmatrix},\ L = \begin{bmatrix} 0_{9\times 3} \\ L_2 \end{bmatrix},\ \text{and}\ H = \begin{bmatrix} H_3 \\ H_4 \end{bmatrix}. \quad (63)$$

I. Accelerometer and Gyroscope As Measurement (MMM) [12, 13]

In this final approach, both accelerometer and gyroscope data are used as measurement inputs, hence, both acceleration and angular velocity appear in the state vector. EKF process and measurement variables are:

$$X = \begin{bmatrix} s \\ v \\ a \\ q \\ \omega \end{bmatrix},\ y = \begin{bmatrix} \mu \\ \gamma \\ \beta \end{bmatrix},\ \rho = \begin{bmatrix} \varepsilon_a \\ \varepsilon_\omega \end{bmatrix},\ \eta = \begin{bmatrix} \varepsilon_\mu \\ \varepsilon_\gamma \\ \varepsilon_\beta \end{bmatrix}. \quad (64)$$

The process equations are given as follows:

$$\begin{aligned} s_t &= s_{t-1} + Tv_{t-1} + \tfrac{1}{2}T^2a_{t-1} + \tfrac{1}{2}T^2\varepsilon_{a,t}, \\ v_t &= v_{t-1} + Ta_{t-1} + T\varepsilon_{a,t}, \\ a_t &= a_{t-1} + \varepsilon_{a,t}, \\ \delta_{\theta,t} &= T\omega_{t-1} + T\varepsilon_{\omega,t}, \\ \omega_t &= \omega_{t-1} + \varepsilon_{\omega,t}. \end{aligned} \quad (65)$$

In addition to camera measurement equation, there are two IMU measurement equations, one involving accelerometer data and the other one involving gyroscope data:

$$\begin{aligned} \gamma_t &= R_t(a_t + g) + \varepsilon_{\gamma,t}, \\ \beta_t &= \omega_t + \varepsilon_{\beta,t}. \end{aligned} \quad (66)$$

Jacobian matrices for this case are calculated as follows:

$$F = \begin{bmatrix} F_3 & 0_{9\times 7} \\ 0_{7\times 9} & F_5 \end{bmatrix},\ V = \begin{bmatrix} V_4 & 0_{9\times 3} \\ 0_{7\times 3} & V_2' \end{bmatrix},\ \text{and}\ H = \begin{bmatrix} H_7 \\ H_8 \\ H_9 \end{bmatrix} \quad (67)$$

where

$$\begin{aligned} H_7 &= H_1Q[-R \quad 0_{3\times 6} \quad H_2(\kappa - s) \quad 0_{3\times 3}], \\ H_8 &= [0_{3\times 6} \quad R \quad H_2(a + g) \quad 0_{3\times 3}], \\ H_9 &= [0_{3\times 13} \quad I]. \end{aligned} \quad (68)$$

This case does not involve control noise Jacobian matrix L as it does not employ a control input.

IV. Synthetic Data Generation

This paper's goal is to compare the different approaches of information fusion in an EKF for tracking the ego-motion of a system composed of a camera and an IMU unit. In order to be able achieve this goal, a systematic way of generating random, yet realistic, motions and corresponding IMU and camera measurements is needed. In this section, we will describe how we generated the data for the simulations and how we used them to compare nine different tracking approaches listed above, under varying motion speeds.

The first task is to generate random "paths" of translation and rotation, which the system undergoes. Higher order derivatives of the paths will be needed to generate IMU data, thus, it is preferable to make such random paths analytical functions of time. The system motion must be realistic, thus, making a simplifying assumption such as circular motion, just for the sake of differentiability weakens the simulation results since the real-life motions cover a lot of motion patterns that are not represented well with such assumptions.

In the light of above considerations, we first randomly choose within a rectangular prism, n waypoints that the system is supposed to pass through. Then we fit cubic splines to these points. The splines are assumed to represent the position of the camera in the World FoR as a function of time. Since cubic splines are analytic functions, higher order

derivatives are readily available. If the total simulation period is T seconds, the camera passes thorough the waypoints at time instants uniformly spaced in $[0, T]$. Thus, depending on the length of the splines between different waypoints, the motion can become faster or slower throughout one path. Figure 2 below shows one sample translational path and corresponding accelerations at sample points computed by taking second derivatives of the fitted spline.

Similar to translation, we also generated random 3D splines that represent rotational motion of the camera-IMU setup. These splines are assumed to correspond to the spherical angles of the orientation of the camera in the Camera FoR. We chose spherical angles instead of another common choice of yaw, pitch and roll angles, since the conversion from the spherical angles to quaternions and rotation matrices are independent of the order of rotations, which is not the case for yaw, pitch and roll angles.

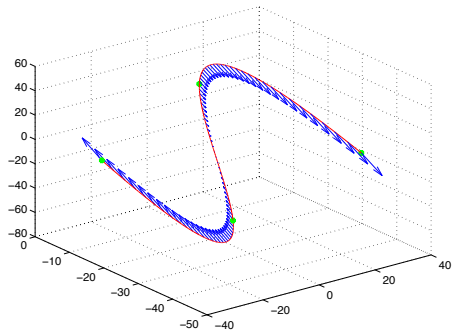

Figure 2. A sample translational path that camera-IMU system undertakes. The red line denotes the path that is a cubic spline fitted to waypoints, which are shown by green dots. The blue arrows denote the accelerations computed from the second derivative of the spline.

From the spherical angles $\theta(t)$, $\varsigma(t)$ and $\psi(t)$ generated above, the quaternion that represents the rotation from the World FoR to the Camera FoR at time instant t can be calculated as follows

$$q_{cw}^{T}(t) = \left[\cos\frac{\theta}{2} \quad \sin\frac{\theta}{2}\cos\varsigma \quad \sin\frac{\theta}{2}\sin\varsigma\cos\psi \quad \sin\frac{\theta}{2}\sin\varsigma\sin\psi\right]. \tag{69}$$

Note that, this quaternion is defined only for the generation of the simulation data, and is not the same as the quaternion in the state vector in EKF equations above.

From the quaternion above, the rotation matrix from the World FoR to the Camera FoR $R_{cw}(t)$ can be calculated using (10). Similarly, since analytical expressions for $\theta(t)$, $\varsigma(t)$ and $\psi(t)$ are available, it is possible to compute $\dot{R}_{cw}(t)$ by differentiating (10) and (69). Then we utilize the following identity [17]

$$\dot{R}_{AB} = [\omega_A]_\times R_{AB} \tag{70}$$

to obtain

$$[\omega_c]_\times = \dot{R}_{cw} R_{cw}^{T}. \tag{71}$$

Above, ω_c denotes the angular velocity of the camera in Camera FoR. From this, the gyroscope measurements become

$$\beta_t = Q^T \omega_c(t) + \varepsilon_{\beta,t}, \tag{72}$$

where Q is the rotation matrix from the IMU FoR to the Camera FoR. $\varepsilon_{\beta,t}$ is a zero mean Gaussian distributed noise component with covariance matrix $\sigma_\beta^2 I$.

Let the position of the camera in World FoR be denoted by $s_w(t) \in \mathbb{R}^3$. To reach the accelerometer measurements, we use the following relation [18]

$$R_{sw}(\ddot{s}_w + g_w) - \xi_s = \omega_s \times \omega_s \times \tau + \dot{\omega}_s \times \tau, \tag{73}$$

where, ω_s is the angular velocity of the IMU in the IMU FoR, and it is given by $Q^T\omega_c$. τ is the IMU to camera displacement in IMU FoR., $\dot{\omega}_s$ can be computed by differentiating (71). g_w is the gravity in the World FoR. ξ_s is the acceleration of the IMU in the IMU FoR. This component is computed from (73), and the accelerometer measurements are obtained as

$$\gamma_{s,t} = \xi_s(t) + \varepsilon_{\gamma,t}, \tag{74}$$

where $\varepsilon_{\gamma,t}$ is a zero mean Gaussian distributed noise component with covariance matrix $\sigma_\gamma^2 I$.

To generate camera readings, we first generated M feature points randomly within a shell of inner radius R_{in} and outer radius R_{out} around the path that the camera-IMU system traverses (Figure 3). Then we used the pinhole model given in (13) to generate the camera measurements. Only those features within the field-of-view of the camera are considered. To achieve this, we assumed an image sensor width im_w and height im_h and only those features that generate measurements within this width and height are kept.

Since the motion blur due to faster motion results in higher detection error, such features should incur higher measurement noise. We obtained this effect as follows. For each feature point, we observed its amount of motion with respect to the previous frame. Let $d_{1,t}$ be the motion in pixels in the first image dimension and $d_{2,t}$ be the one in the second dimension. Then the variances of the components of the zero mean Gaussian random variables $\varepsilon_{\mu,t}$ (*i.e.*, the camera measurement noise) are assumed to be

$$\sigma_{\mu_i,t}^2 = \sigma_c^2 + \alpha d_{i,t}^2. \tag{75}$$

This way, an increased amount of noise is added to fast moving feature point measurements.

V. Simulation Results

We generated random data as explained in Section IV for 110 different simulation runs, each for a duration of 500 camera frames, which amounts to 33.33 seconds since the camera is assumed to run at 15 frames per second. The IMU unit is assumed to run at 120 Hz, thus, 4000 accelerometer and gyroscope measurements per run are also computed. The dimensions of the rectangular prisms, inside of which the translational and angular waypoints are chosen, are 100 cm and 0.2π. We assumed $n = 4$ waypoints for translation and another 4 for rotation, however, in general, any number is possible using multiple connected splines.

The inner radius R_{in} of the shell in which features are placed is 200 cm, and the outer radius R_{out} is 300 cm. We placed $M = 500$ features in the shell randomly for each simulation. Noise variance parameters are $\sigma_c = 1$ pixel, $\alpha = 0.2$, $\sigma_\gamma = 10^{-3}$ cm/s^2 and $\sigma_\beta = 10^{-4}$ rad/s. The focal length of the camera is assumed to be 700 pixels, while the imager width and height are assumed to be 640 and 480 pixels respectively, resulting in about 49° horizontal and 38° vertical field of view. Without loss of generality, we assumed $Q = I$ and $\tau = 0$.

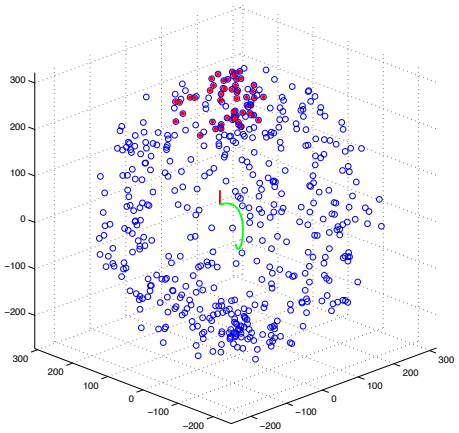

Figure 3. A snapshot of the camera measurement generation process. The red stick represents the camera, the green curve represents the path that the camera follows. The little hollow circles represent all the feature points in the 3D map, whereas the filled purple circles are the feature points that are in the current field-of-view of the camera.

In the simulations, for the process noise standard deviations, we assumed $\sigma_\omega = 0.1$ rad/s (the standard deviation of $\varepsilon_{\omega,t}$) and $\sigma_v = 0.15$ cm/s (the standard deviation of $\varepsilon_{v,t}$). The standard deviation of $\varepsilon_{\theta,t}$ is assumed $\sigma_\omega T_s$ and that of $\varepsilon_{a,t}$ is assumed σ_v / T_s, where T_s is the simulation step size, and equal to 1/(120 Hz) = 0.0083 s. The EKF results do change with this choice. We first ran initial tests with several process noise standard deviation values and decided that these values give reasonable results. In reality, similar to our approach, one can first do these tests in simulation with motion patterns representative of real life, decide on the process noise standard deviation values and use them in real experiments. For different motion speeds, the process noise standard deviations are increased/decreased in proportion to the speed.

For the simulations, first, 110 paths and corresponding camera and IMU data are generated and stored. The different flavors of EKF trackers are run on the same stored data and their 3D position and 3D orientation RMSEs are reported. The re-projection RMSEs are also calculated and in order to exclude possible outliers, the 10 runs with biggest re-projection errors are thrown away and all RMSEs are calculated with the remaining 100.

In this paper, another goal is to test the tracker flavors under different motion speeds. However, for a fair comparison, we would like to use similar motion patterns to the original 110, but just faster or slower. To achieve this, we multiplied the splines' coordinates by 2 (for faster) or 0.5 (for slower), without changing the time instants. Thus, a data set of 110 runs for the "fast" and another set of 110 runs for the "slow" motion case are obtained. The RMSEs for these cases are also calculated as explained above.

In Figure 4, we report the RMSE results for fast, default and slow motion speeds and different tracker cases. The leftmost column has the RMSE in the tracked position of the IMU, the middle column has the RMSE of the tracked quaternion (as a measure of the orientation RMSE), and the rightmost column has the RMSE in pixels in the re-projection of the feature points using the tracked camera pose. The results in Figure 4 reveal the following:

i) As explained in the introduction, our extensive simulations confirm that the IMU information is useful in improving the tracking performance, and the benefits become more pronounced as the motion becomes faster.

ii) Our extensive simulations suggest that it is better to use IMU data as measurement as opposed to control input; hence the best combination is the MMM case, across all speeds. It appears that the linearization approximation at the correction step does not effect the accuracy of the tracker to the point where using the measurements at the prediction step as control inputs is preferred. Furthermore, since gyroscope measurement equations are already linear in the state variables, the benefit of using IMU data as measurement is more pronounced for the gyroscope.

iii) In general, the accelerometer data helps improve the position accuracy more than the orientation, whereas the gyroscope information helps improve the orientation accuracy more than the position. This makes sense, since accelerometer measures translational motion whereas the gyroscope measures rotational motion.

iv) Gyroscope appears to have more influence in reducing the projection error than accelerometer. Errors in the 3D camera position translate to projection errors that diminish with the distance to the feature points in the scene. On the other hand, the projection errors due to camera orientation errors do not change with distance. Thus, the average projection error over near and distant feature points is more prone to camera orientation errors, which is improved most by the gyroscope data as explained in (iii).

v) The improvement provided by gyroscope in position is more pronounced than the improvement provided by accelerometer in orientation. As explained in (iv), the gyroscope data has a bigger effect on the projection errors than the accelerometer data. During tracking, projection errors induce errors in the camera measurement innovations at the correction step of the EKF. This translates to errors in position, explaining the greater effect of gyroscope measurements to position accuracy. Hence, if only one inertial sensor is to be used, it should be gyroscope used as measurement.

vi) For the simulation settings, it is possible to achieve less than 2 pixels RMS re-projection error with the MMM EKF (both inertial sensors used as measurement) even under fast translational and rotational motions.

VI. Experimental Results with Real Data

Results presented in the previous section provide a thorough comparison of different fusion approaches in a realistic simulation setting. In this section, we present experimental results with real data that corroborate the above simulation results. In order to collect visual and inertial sensor data we assembled a hand-held capture unit using a FIREFLY-FFMV-03M2C camera and an STEVAL-MKI062V2 inertial sensor unit. The inertial sensor unit houses a triple-axis accelerometer and a triple-axis gyroscope which were both run at a rate of 120 Hz. The camera was adjusted to capture 1600x1200 resolution frames at a rate of 30 Hz. We recorded 10 seconds of synchronous video and inertial sensor data. The video data was processed using the SIFTGPU software [19] to extract and match 2D image features, and the BUNDLER software [20] to obtain a 3D map of the scene as well as the 3D pose sequence (Figure 5) of the camera during the 10 seconds. The 2D features and the 3D map were used during EKF tracking, and the 3D pose sequence is used as the ground truth data to evaluate the RMSE performance of different EKF tracking schemes. Figure 6 shows the resulting RMSE values, which confirm that (i) both sensors help improve the tracking accuracy more when used as measurements as opposed to control inputs and (ii) accelerometer helps more with the 3D position accuracy while gyroscope helps more with the 3D orientation accuracy.

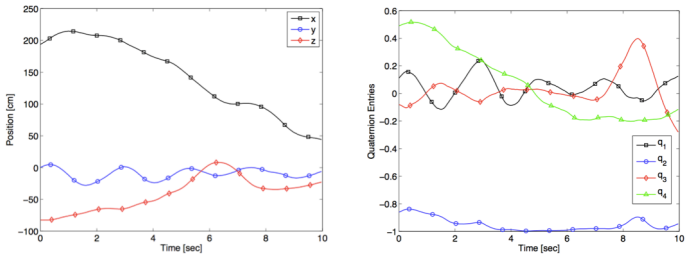

Figure 5. 3D motion of the camera calculated using the BUNDLER software and assumed to be the ground truth for obtaining the RMSE values (left: x-y-z components of the position, right: quaternion components of the orientation).

VII. Conclusion

In this paper, we provide a thorough analysis of different approaches of fusing accelerometer and gyroscope data to camera measurements in EKF. We compare all eight possible combinations of using inertial sensor data as control or measurement inputs, and the camera-only case to provide a baseline for comparisons, via extensive and realistic simulations using the same data set collected at different motion speeds, as well as real data. Three of these cases, namely, (i) gyroscope only as control, (ii) gyroscope as measurement and accelerometer as control, and (iii) accelerometer as measurement and gyroscope as control were not covered in the literature before.

We provide a complete set of EKF equations including the Jacobian matrices employed by EKF for all eight fusion cases and the camera only case. Our major finding is that both inertial sensors improve 3D accuracy more when used as measurement inputs, hence the best combination is the MMM case, across all speeds. Furthermore, we find that the improvement provided by gyroscope in position is more pronounced than the improvement provided by accelerometer in orientation, hence if only one inertial sensor is to be used, it should be gyroscope used as measurement.

Previous to our work, the most extensive study of fusing inertial sensor data at the EKF has been conducted in [12], where MXM, MCC, and MMM cases have been compared and it is concluded that MMM and MCC exhibit similar performance and both provide better tracking accuracy than MXM at fast motion speeds. However, with the confidence of our extensive simulations we believe that the correct ordering among these three cases should be MMM, MXM, and MCC, at all speeds, since using gyroscope only as measurement has more effect on the tracking performance than using both sensors as control inputs as explained in the simulations results section.

Another important contribution of our work is a 3D spline based generation of realistic synthetic data corresponding to gyroscope, accelerometer and camera measurements. This simulation environment becomes instrumental in extensive testing of the EKF and other possible trackers under real-life motion patterns, assess their performance, as well as observe and compare with ground-truth the tracker's inner workings and states.

References

[1] R. T. Azuma, "A Survey of Augmented Reality," Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355-385, Aug.1997.

[2] A. Schmeil and W. Broll, "Mara: an augmented personal assistant and companion," *ACM SIGGRAPH 2006 Sketches*, New York, NY, USA, 2006, p. 141.

[3] G. Papagiannakis, G. Singh, N. Magnenat-Thalmann, "A survey of mobile and wireless technologies for augmented reality systems," *Journal of Computer Animation and Virtual Worlds*, vol. 19, no. 1, pp. 3-22, Feb. 2008.

[4] G.Papagiannakis, N. M. Thalmann, "Mobile Augmented Heritage: Enabling Human Life in ancient Pompeii," *International Journal of Architectural Computing*, vol. 2, no. 5, pp. 395-415, July 2007.

[5] Y. Bar-Shalom, X. Rong Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*, John Wiley & Sons, 2004.

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

[7] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre, "Recent Advances in Augmented Reality," *IEEE Computer Graphics and Applications*, 2001.

[8] L. Armesto, J. Tornero, M. Vincze, "Fast ego-motion estimation with multi-rate fusion of inertial and vision," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 577-588, June 2007.

[9] P. Gemeiner, P. Einramhof, M. Vincze, "Simultaneous motion and structure estimation by fusion of inertial and vision data," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 591-605, June 2007.

[10] D. Strelow, "Motion estimation from image and inertial measurements," Ph.D. dissertation, CMU, 2004.

[11] Y. Yokokohji, Y. Sugawara, and T. Yoshikawa, "Accurate Image Overlay on Video See-through HMDs Using Vision and Accelerometers," *Proc. IEEE Virtual Reality*, Los Alamitos, Calif., pp. 247-254, 2000.

[12] G. Bleser, D. Stricker, "Advanced tracking through efficient image processing and visual- inertial sensor fusion," *Computers & Graphics*, vol. 33, no. 1, pp. 59-72, Feb. 2009.

[13] J. Chen, W. Liu, Y. Wnag, J. Guo, "Fusion of inertial and vision data for accurate tracking," *SPIE Int. Conference on Machine Vision*, Jan. 2012.

[14] A. O. Ercan and A. T. Erdem, "On Sensor Fusion for Head Tracking in Augmented Reality Applications," *IEEE American Control Conference*, pp. 1286-1291, 2011.
[15] R. Szeliski, *Computer Vision: Algorithms and Appl.*, Springer, 2011.
[16] N. Snavely, S. M. Seitz, R. Szeliski, "Modeling the World from Internet Photo Collections," *International Journal of Computer Vision*, 2007.
[17] F. Lizarralde, J. T. Wen, "Attitude Control without Angular Velocity Measurement: A Passivity Approach," *IEEE Int. Conf. Robotics and Automation*, pp. 2701-2706, 1995.
[18] A. T. Erdem, A. O. Ercan, T. Aydin, "Internal calibration of a camera and an inertial measurement unit," *Signal Processing and Communications Applications Conference (SIU)*, pp. 24-26, April 2013.
[19] C. Wu, "Siftgpu: A gpu implementation of scale invariant feature transform," http://cs.unc.edu/~ccwu/siftgpu/, 2013.
[20] Noah Snavely, Steven M. Seitz, Richard Szeliski. "Photo Tourism: Exploring image collections in 3D," ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006), 2006

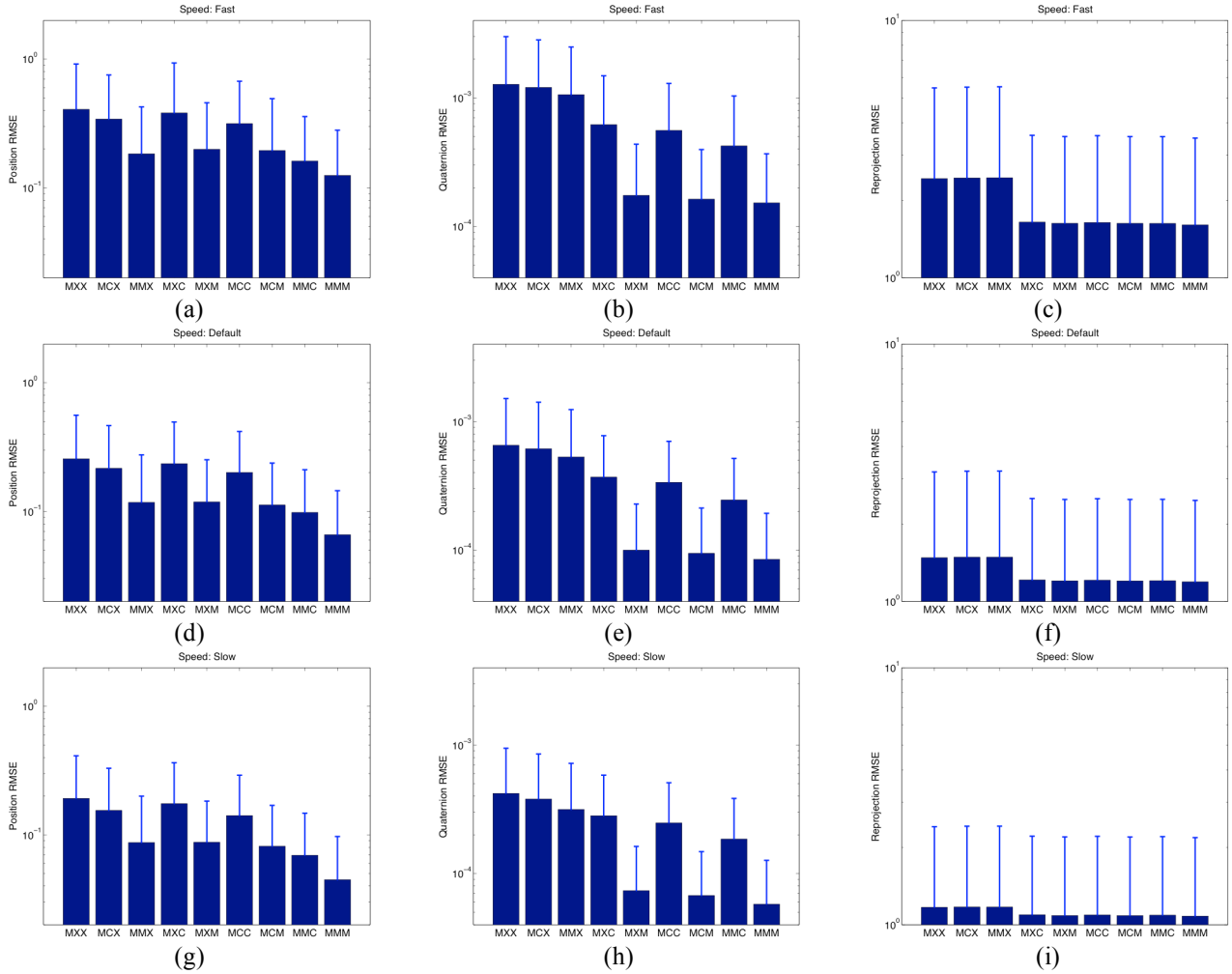

Figure 4. RMSE results for different motion speeds and different tracker cases. Performances of different trackers for fast, default and slow motion speeds are given in sub-figures (a)–(c), (d)–(f) and (g)–(i), respectively. The leftmost column has the RMSE in the tracked position of the IMU, the middle column has the RMSE of the tracked quaternion (as a measure of the orientation RMSE), and the rightmost column has the RMSE in pixels in the reprojection of the feature points using the tracked camera pose. The thick solid bars represent the means of 100 simulation runs and the thin stick error bars represent one standard deviation.

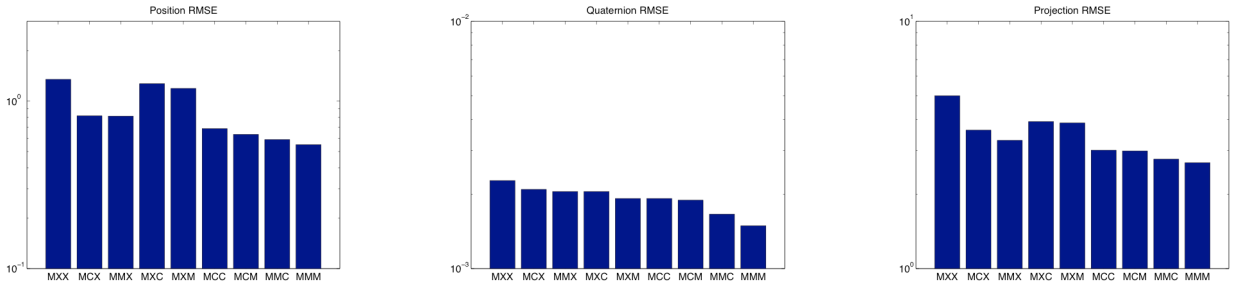

Figure 6. Results of experiments with real data: (a) Position, (b) orientation, and (c) projection RMSE values for different tracker cases.

A. Tanju Erdem (S'88-M'91) received a B.S. degree in electrical and electronics engineering and a B.S. degree in physics from Bogazici University, Istanbul, in 1986. He received the M.S. and Ph.D. degrees in electrical engineering from University of Rochester, Rochester, NY, in 1988 and 1990, respectively. He was with the Research Laboratories of Eastman Kodak Company, Rochester, NY, from 1990 to 1998. He held a part-time faculty position at the University of Rochester during the same time. He was a visiting faculty at Bilkent University, Ankara, in 1996. He co-founded Momentum Digital Media Technologies, Inc., in 1998 in Istanbul, and served as its CTO until 2009. He joined Ozyegin University in 2009, where he served as the Chair of Computer Science Department in 2013 and has been serving as the Dean of Engineering School since 2014.

He was a member of the MPEG Committee from 1992 to 1998, where he served as the Chairman of the MPEG-2 Ad-Hoc Group on 10-bit Video in 1993. He was an Area Editor for Signal Processing: Image Communication between 2009-2014. He served as the President of Rochester Chapter of IEEE Signal Processing Society between 1992–1994. He was the general Co-Chair of the 20th Signal Processing and Communication Applications Conference (SIU) in 2012. His research interests are in the areas of 3D computer vision, video analytics, 3D animation, video game development, and game based learning. He serves as an independent expert to review projects for the European Commission. He holds eight US patents.

Ali Ozer Ercan (S'02-M'08) received his B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey in 2000, and his M.S. and Ph.D. degrees in electrical engineering from Stanford University, CA, USA in 2002 and 2007 respectively. From 2007 until 2009, he was with the Berkeley Wireless Research Center of University of California, Berkeley, CA, USA, for post-doctoral studies. He joined Özyeğin University, Istanbul, Turkey, in September 2009, where he is currently an Assistant Professor. He received FP7 Marie Curie International Reintegration Grant in 2009. He served as the Publications Chair of IEEE 3DTV Conference (3DTV-CON) in 2011, and Technical Program Co-chair and Publications Chair of IEEE Signal Processing and Communication Applications Conference (SIU) in 2012. His research interests are in the areas of signal and image processing, computer vision, and wireless communication networks.