2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Smartphones Power Flying Robots

Giuseppe Loianno[†], Yash Mulgaonkar[†], Chris Brunner[*], Dheeraj Ahuja[*], Arvind Ramanandan[*], Murali Chari[*], Serafin Diaz[*], and Vijay Kumar[†]

***Abstract*— Consumer grade technology seen in cameras and phones has led to the price/performance ratio of sensors and processors falling dramatically over the last decade. In particular, most devices are packaged with a camera, a gyroscope, and an accelerometer, important sensors for aerial robotics. The low mass and small form factor make them particularly well suited for autonomous flight with small flying robots, especially in GPS-denied environments. In this work, we present the first fully autonomous smartphone-based quadrotor. All the computation, sensing and control runs on an off-the-shelf smartphone, with all the software functionality in a smartphone app. We show how quadrotors can be stabilized and controlled to achieve autonomous flight in indoor buildings with application to smart homes, search and rescue, construction and architecture. The work allows any consumer with a smartphone to autonomously drive a quadrotor robot platform, even without GPS, by downloading an app, and concurrently build 3-D maps.**

I. Introduction

Micro Aerial Vehicles (MAVs) equipped with on-board sensors, are ideal platforms for autonomous navigation in complex and confined environments for solving tasks such as exploration [1], inspection [2], mapping, interaction with the environment [3], [4] and, search and rescue [5]. While MAVs are now available as off-the shelf products from such companies as DJI[1], 3D Robotics[2], Parrot[3] and Yuneec[4], most products rely on GPS, with vision and acoustic sensors used only for altitude stabilization.

However, recent research on MAVs has yielded a number of significant results. There are a number of Simultaneous Localization and Mapping (SLAM) approaches for MAVs. Good results have been obtained using monocular cameras and IMUs (Inertial Measurements Units consisting of gyroscope and accelerometer) [6], [7], stereo camera configurations [1], [8], [9], and RGB-D sensor systems [5], [10]. In [5], [10], a Kinect and the vehicles onboard sensors are used to perform state estimation using an EKF, while in [6], the same filter is used to combine monocular visual information with inertial sensor data to solve the scale factor problem. However, these algorithms generally only work without limitation on laptop or desktop computers. Moreover, RGB-D sensors have low quality cameras and suffer during exposure to direct sunlight. The minimal sensor suite for autonomous localization consists of two inexpensive, lightweight and widely available sensors, a single camera and an IMU as shown in [11]–[14] including theoretical observability analysis and applied to aerial navigation in [6], [7]. However, most solutions are heavy and over 1 kg in mass. Second, there is still a gap between the use of complex algorithms in the research field and its use by naive humans for everyday applications.

Fig. 1: The flying vehicle demonstrated during CES 2015 in Las Vegas.

In this paper, we leverage Commercial Off-The-Shelf (COTS) consumer-grade products like smartphones to overcome these two shortcomings. As the number of smartphones has grown to be larger than the world population[5], the integrated sensing (e.g. gyroscopes, accelerometers, and high resolution cameras) and computation, makes it possible to use them for low-cost brains and sensors for robots. This article addresses the use of an off-the-shelf quadrotor platform with a COTS smartphone (see Fig. 1). Current algorithms, based on cameras and Inertial Measurement Units (IMUs), need a large platform and customized hardware processors that are heavy and expensive. Our work addresses this gap and demonstrates that a COTS smartphone (in this case, a Samsung Galaxy S5) is able to provide full autonomy for an aerial robotic platform. The smartphone incorporates, embedded in an app, software for control, planning and to track the full 3-dimensional motion, while concurrently creating a map of the environment using visual odometry and structure from motion algorithms.

[†]The authors are with the GRASP Lab, University of Pennsylvania, 3330 Walnut Street, 19103 Phialdelphia, USA. email: {loiannog, yashm, kumar}@seas.upenn.edu.

[*]The authors are with Qualcomm Technologies, Inc., 5775 Morehouse Drive, San Diego, {chris, dahuja, rarvind, mchari, sdiaz}@qti.qualcomm.com

[1]http://http://www.dji.com/

[2]http://www.3drobotics.com/

[3]http://www.parrot.com/

[4]http://www.yuneec.com/

[5]http://www.theinquirer.net/inquirer/news/2374525/there-are-now-more-active-mobile-devices-than-humans

Relevant to this paper, are previous works on implementing vision based algorithms on camera phones. However, these algorithms rely on marker tracking [15] and localization algorithms [16] which are suitable for augmented reality applications. However, they are not suitable to deal with long term operations and large navigation coverage areas, needed in robotic tasks. Good results on camera phones have been obtained considering rolling shutter effects [17]. However, the image were collected at a slower rate (15 Hz) and processed later.

In our previous work [18], a first attempt toward autonomous flight based on a phone is presented. It is worth noting that, the phone was a special device Google Tango[6], which incorporates an enhanced RGB-D sensor with a global shutter camera, a fisheye lens that provides a field of view of 170°, a high quality IMU and a depth sensor able to capture a dense set of point clouds. This set of specialized sensors allows the extraction of depth and solves the scale problem that affects monocular visual odometry algorithms. Finally, the state estimation, control and planning was done on a second independent Odroid-XU[7] board, with additional onboard vehicle sensors.

In this work, we present the complete architecture, the algorithms and the software for automating aerial robot flight with a camera phone. It represents the first "plug and play" integration of a consumer product with an off-the-shelf aerial robot to enable autonomy with possible onboard localization, mapping and control. Thus, it is representative a new class of affordable smart devices that can potentially lower the barrier to automation into homes by providing services for localization, state estimation, control and mapping. Any end user may be able to utilize his smartphone to autonomously control an aerial platform and to add new functionalities.

This work presents multiple contributions. First, we developed a quadrotor platform equipped with a COTS smartphone. Any user can build his own quadrotor and autonomously fly it with its own phone device. Second, a state estimator based on visual inertial odometry (VIO) provides the vehicle's pose at 200 Hz, enabling fast motions. Then, a non-linear controller also running on the phone is able to guarantee the exponential stability of the platform enabling trajectory following in 3D space. Finally, this is the first time that a sophisticated platform, like a quadrotor, has been controlled by a consumer device. All the software components, which include planning, control, estimation and mapping are running on a phone embedded in a single app.

The paper is organized as follows. In Section II, a general overview of our hardware and software frameworks is presented. In Section III, the dynamics of the quadrotor and the control framework are provided, whereas in Section IV, the strategy to obtain the pose of the vehicle at high rate, enabling autonomous flight, is shown. Section V presents extensive results on autonomous stabilization and navigation with the proposed prototype. Section VI concludes the work and provides an overview of the multiple future scenarios that can be enabled by our technology in the future.

[6]https://www.google.com/atap/projecttango

[7]http://www.hardkernel.com

II. System Architecture

Our platform of choice was a quadrotor due to its mechanical simplicity [19] and ease of control. Moreover, its ability to operate in confined spaces, hover in space and perch or land on a flat surfaces makes it a very attractive aerial platform with tremendous potential. In the following, a brief description of the proposed hardware and software architecture is presented. A schematic of the proposed approach is depicted in Fig. 2.

A. Hardware Architecture

The experimental platform shown in Fig. 1 is made from carbon fiber COTS components and is equipped with 4 brushless motors and an autopilot board consisting of an IMU and a user–programmable ARM7 microcontroller. The only other addition to this setup is a forward–pointing smartphone Samsung S5, and an USB-serial cable to deal with the communication between the smartphone and the microcontroller. The total mass of the platform is 750 g. It should be noticed that our experimental setup is independent of the specifics of the employed phone. In particular, the system we are proposing just uses 50% of the total CPU phone power giving the possibility to execute all the tasks on a less powerful phone device without limitations and add other functionalities in the future.

B. Software Architecture

The main software components are a position and attitude controller (see light blue box in Fig.2), a state estimation algorithm composed of VIO described in Section IV, which processes images at 30 Hz and an Unscented Kalman Filter (UKF) to deal with control constraints for fast motions (see green box in Fig.2). The control receives the estimated pose at 200 Hz from the UKF and sends the attitude commands to a microcontroller on the vehicle. To provide an interaction with the vehicle, two interfaces have been developed. The first one is ROS[8] based and can run on a common laptop. The second one is more appealing for naive users and it is another app running on a remote android device like phone and tablet. The interface is able to show the reconstructed environment, the estimated 3D path, send high level commands, and change control and flight parameters settings. The communication has been realized via UDP and it runs in a separate thread to not affect estimation and control. All the tasks are then executed on the phone providing state estimation and control at a fixed rate of 200 Hz. The estimation, control and planning have been realized in separate threads to guarantee real-time reliability. It should be pointed out that the presented strategy allows the vehicle to run all algorithms on board. The base station is responsible only for visualization, as well as handling user interaction. All the components are explained in detail in the following sections.

[8]www.ros.org

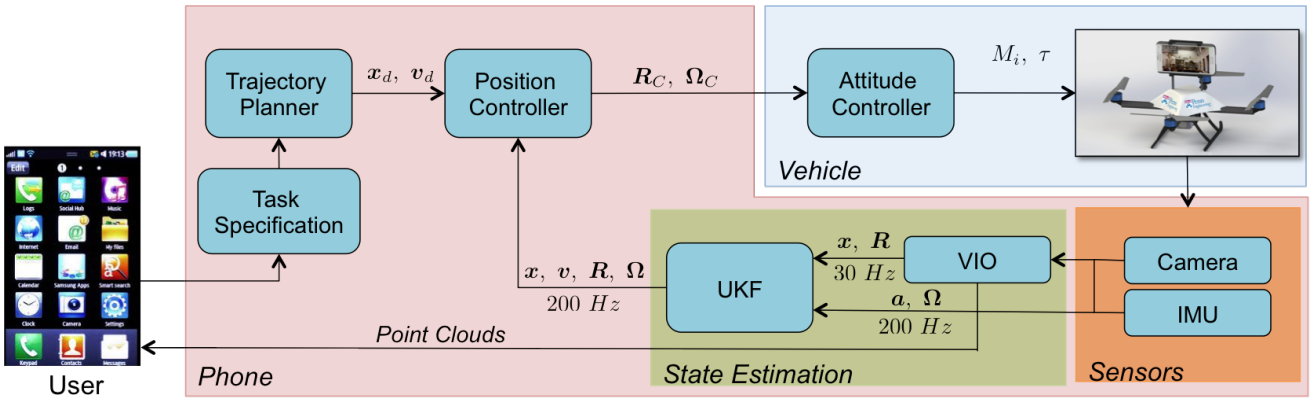

Fig. 2: The proposed architecture.

III. System Model

A quadrotor is a system made of four identical rotors and propellers located at the vertices of a square. The first and third propeller rotate clockwise, the second and fourth propeller rotate counterclockwise (see Fig. 3).

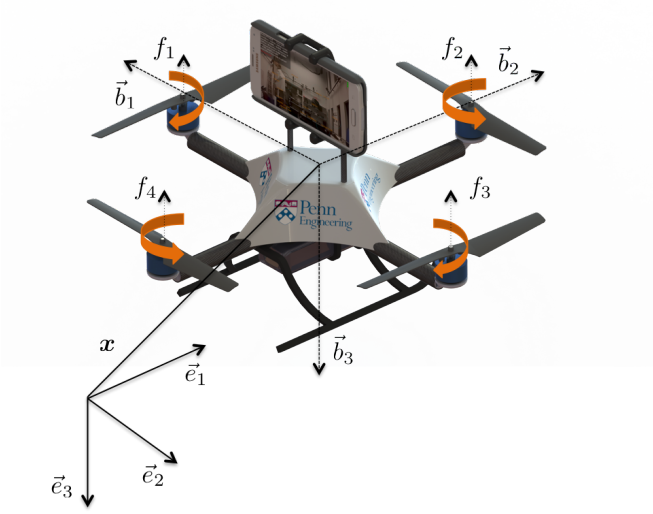

Fig. 3: The quadrotor CAD model and the corresponding reference frames.

A. Dynamic Model

Let us consider an inertial reference frame denoted by $\begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3\end{bmatrix}$ and a body reference frame centered in the center of mass of the vehicle denoted by $\boldsymbol{R} = \begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3\end{bmatrix}$ where $\boldsymbol{R} \in SO(3)$. The dynamic model of the vehicle can be expressed as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{v}, \\ m\dot{\boldsymbol{v}} &= -\boldsymbol{R}\tau\boldsymbol{e}_3 + mg\boldsymbol{e}_3, \\ \dot{\boldsymbol{R}} &= \boldsymbol{R}\hat{\boldsymbol{\Omega}}, \\ \boldsymbol{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega} &= \boldsymbol{M}, \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^3$ is the Cartesian position of the vehicle expressed in the inertial frame, $\boldsymbol{v} \in \mathbb{R}^3$ is the velocity of the vehicle in the inertial frame, $m \in \mathbb{R}$ is the mass, $\boldsymbol{\Omega} \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame and $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the inertia matrix with respect to the body frame. The hat symbol $\hat{}$ denotes the skew-symmetry operator according to $\hat{\boldsymbol{x}}\boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$, g is the standard gravitational acceleration and $\boldsymbol{e}_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$. The total moment $\boldsymbol{M} \in \mathbb{R}^3$, with $\boldsymbol{M} = \begin{bmatrix}M_1 & M_2 & M_3\end{bmatrix}^\top$, along all axes of the body-fixed frame and the thrust $\tau \in \mathbb{R}$ are control inputs of the plant. The dynamics of rotors and propellers are neglected and it is assumed that the force of each propeller is directly controlled. The total thrust, $\tau = \sum_{j=1}^{4} f_j$, acts in the direction of the z axis of the body-fixed frame, which is orthogonal to the plane defined by the centers of the four propellers. The relationship between single motor force f_j, the total thrust τ and the total moment $\boldsymbol{M}$ can be written as

$$\begin{bmatrix}\tau \\ M_1 \\ M_2 \\ M_3\end{bmatrix} = \begin{bmatrix}1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ d & 0 & -d & 0 \\ -c & c & -c & c\end{bmatrix} \begin{bmatrix}f_1 \\ f_2 \\ f_3 \\ f_4\end{bmatrix} \tag{2}$$

where c a constant value and d is the distance from the center of mass to the center of each rotor in the $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ plane. For non-zero values of d, eq. (2) can be inverted. Our assumption that τ and $\boldsymbol{M}$ are the inputs of the plant is therefore valid.

B. Position and Attitude Controllers

In most previous works, a back-stepping approach is used for control because the attitude dynamics can be assumed to be faster than the dynamics governing the position, and linearized controllers are used for both loops [6], [19]. In this work, because we need to model large excursions from the hover position for robustness, we use a nonlinear controller based on the work of [20], [21].

The control inputs τ, M are chosen as

$$\begin{aligned} \boldsymbol{M} = &- k_R\boldsymbol{e}_R - k_\Omega\boldsymbol{e}_\Omega + \boldsymbol{\Omega} \times J\boldsymbol{\Omega} - \\ &\boldsymbol{J}\left(\hat{\boldsymbol{\Omega}}\boldsymbol{R}^\top\boldsymbol{R}_C\boldsymbol{\Omega}_C - \boldsymbol{R}^\top\boldsymbol{R}_C\dot{\boldsymbol{\Omega}}_C\right), \\ \tau = &- \left(-k_x\boldsymbol{e}_x - k_v\boldsymbol{e}_v - mg\boldsymbol{e}_3 + m\ddot{\boldsymbol{x}}_d\right) \cdot \boldsymbol{R}\boldsymbol{e}_3, \end{aligned} \tag{3}$$

with $\ddot{\boldsymbol{x}}_d$ the desired acceleration, k_x, k_v, k_R, k_Ω positive definite terms. The subscript C denotes a commanded value.

The quantities

$$\boldsymbol{e}_R = \frac{1}{2}\left(\boldsymbol{R}_C{}^\top \boldsymbol{R} - \boldsymbol{R}^\top \boldsymbol{R}_C\right)^\vee, \;\; \boldsymbol{e}_\Omega = \boldsymbol{\Omega} - \boldsymbol{R}^\top \boldsymbol{R}_C \boldsymbol{\Omega}_C, \\ \boldsymbol{e}_x = \boldsymbol{x} - \boldsymbol{x}_d, \;\; \boldsymbol{e}_v = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d, \tag{4}$$

represent the orientation, angular rate errors, and translation errors respectively. The symbol $.^\vee$ represent the *vee* map $\mathfrak{so}(3) \to R^3$. If the initial attitude error is less than $90°$, the zero equilibrium of the tracking errors is exponentially stable, i.e., $\left[\boldsymbol{e}_x^\top \;\; \boldsymbol{e}_v^\top \;\; \boldsymbol{e}_\Omega^\top \;\; \boldsymbol{e}_R^\top\right]^\top \equiv \left[\mathbf{0}^\top \;\; \mathbf{0}^\top \;\; \mathbf{0}^\top \;\; \mathbf{0}^\top\right]^\top$. Further, if the initial attitude error is between $90°$ and $180°$, then the zero equilibrium of the tracking errors is almost globally exponentially attractive. The reader can refer to [20] for convergence and stability analysis and to [21] for experimental results.

IV. State Estimation

In this section, we will describe the different steps that enable to recover the 6-DOF (Degree Of Freedom) pose of the vehicle in the inertial frame defined in Section III, necessary to control the vehicle. The subscripts c, b, and s denote the camera, body, and inertial frame, respectively, while the subscripts a, g denote accelerometer and gyros quantities. Without loss of generality, in the following we will suppose that the body frame defined in Section III is coincident with the IMU frame. The symbol $\boldsymbol{R}_{bc}$ denotes the orientation of frame c with respect to frame b.

A. Visual Inertial Odometry

The goal of the VIO system is to localize the dynamics of the body with respect to the inertial frame using accelerometers and gyroscopes as interoceptive and cameras as exteroceptive sensors. The navigation state vector $\mathbf{x}(t) \in \mathbb{R}^{12} \times \boldsymbol{se(3)}$ is defined as

$$\mathbf{x} = \left[\begin{array}{ccccccc} \boldsymbol{x}_{sb}^\top & \boldsymbol{\Theta}_{sb}^\top & \boldsymbol{v}_{sb}^\top & \boldsymbol{\gamma}^\top & \boldsymbol{b}_g^\top & \boldsymbol{b}_a^\top \end{array}\right]^\top, \tag{5}$$

where $\boldsymbol{x}_{sb} \in \mathbb{R}^3$ denotes the vector from the origin of the inertial frame to the origin of the b frame expressed in the inertial frame, $\boldsymbol{v}_{sb}$ is its time derivative, $\boldsymbol{\Theta}_{sb}$ is the attitude vector in exponential coordinates [22], $\boldsymbol{\gamma}$ is the unknown gravity vector in the inertial frame, and $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ denote slowly changing accelerometer and gyroscope biases. The camera is rigidly mounted to the accelerometer. For brevity, we will consider that the relative transformation between camera and accelerometer is known. However, in our current setup the calibration parameters, such as inertial sensor scale factor, non-orthogonality, camera-accelerometer transformation $(\boldsymbol{x}_{bc}, \boldsymbol{\Theta}_{bc})$ are jointly estimated by appending them to the state. The kinematics of eq. (5) can be derived as

$$\dot{\boldsymbol{x}}_{sb} = \boldsymbol{v}_{sb}, \;\; \dot{\boldsymbol{R}}_{sb} = \boldsymbol{R}_{sb}{}^b\hat{\boldsymbol{\Omega}}_{sb}, \;\; \dot{\boldsymbol{v}}_{sb} = \boldsymbol{a}, \\ \dot{\boldsymbol{\gamma}} = \boldsymbol{n}_\gamma, \;\; \dot{\boldsymbol{b}}_g = \boldsymbol{n}_{bg}, \;\; \dot{\boldsymbol{a}}_b = \boldsymbol{n}_{ba}, \tag{6}$$

where ${}^b\boldsymbol{\Omega}_{sb} = \boldsymbol{R}_{bg}\boldsymbol{\Omega}_{sg}$ is the angular velocity of the body frame relative to the inertial, expressed in the body frame, $(\boldsymbol{n}_{bg}, \boldsymbol{n}_{ba}, \boldsymbol{n}_\gamma)$ denote the random walk parameters for biases and gravity.

The inputs to the dynamic system $\left(\boldsymbol{a}, {}^b\boldsymbol{\Omega}_{sb}\right)$

$$\boldsymbol{a} = \boldsymbol{R}_{sb}\boldsymbol{A}_a^{-1}\left(\boldsymbol{y}_a - \boldsymbol{b}_a - \boldsymbol{n}_a\right) + \boldsymbol{\gamma}, \tag{7}$$

$${}^b\boldsymbol{\Omega}_{sb} = \boldsymbol{R}_{bg}\boldsymbol{A}_g^{-1}\left(\boldsymbol{y}_g - \boldsymbol{b}_g - \boldsymbol{n}_g\right) \tag{8}$$

are derived from the measurements $\boldsymbol{y}_a, \boldsymbol{y}_g \in \mathbb{R}^3$ of the accelerometer and gyroscope. These are modelled as

$$\boldsymbol{y}_a = \boldsymbol{A}_a\boldsymbol{R}_{sb}^\top\left(\boldsymbol{a} - \boldsymbol{\gamma}\right) + \boldsymbol{b}_a + \boldsymbol{n}_a, \tag{9}$$

$$\boldsymbol{y}_g = \boldsymbol{A}_g\boldsymbol{\Omega}_{sg} + \boldsymbol{b}_g + \boldsymbol{n}_g, \tag{10}$$

where $\boldsymbol{a}(t)$ denotes the acceleration of the body relative in the inertial frame, $\boldsymbol{\Omega}_{sg}$ denotes the angular velocity of the gyroscope relative to the inertial frame (expressed in the IMU frame), $\boldsymbol{n}_a$ and $\boldsymbol{n}_g$ denote additive sensor noise and $\boldsymbol{A}_a$, $\boldsymbol{A}_g$ characterize the effects of non-unit scale and non-orthogonality of the sensor axes. Errors in a mechanization of eq. (6) grow in an unbounded fashion[9], necessitating integration of sensor with bounded errors. In the past decade, camera measurements have been effectively used to aid Inertial Navigation Systems [11]–[13], [17]. It is assumed that the world is populated with several distinguishable features that can be easily tracked over time using a camera. If $\boldsymbol{T}_f^\top(t) = \left[\begin{array}{ccc} x & y & z \end{array}\right]$ denotes the vector to the feature in the camera frame at time t, the measurement of the camera (normalizing for its intrinsic parameters) is given by the standard perspective projection model

$$\boldsymbol{y}_c = \frac{1}{z}\left[\begin{array}{c} x \\ y \end{array}\right] + \boldsymbol{n}_c, \tag{11}$$

where $\boldsymbol{n}_c$ is pixel measurement noise.

Consider the scenario depicted in Fig. 4. If a feature f is tracked at each t_i, $i = 0, \ldots 3$, then the camera measurements $\boldsymbol{y}_c(t_i)$ of the feature provide constraints on the states $\mathbf{x}(t_i)$ during that interval. Let $\boldsymbol{F}$ denote the integral function corresponding to eq. (6) such that

$$\mathbf{x}(t_i) = \boldsymbol{F}\left(\mathbf{x}(t_0), \boldsymbol{u}, t_i\right), \tag{12}$$

where $\boldsymbol{u}$ is the set of all inertial measurements in $[t_0, t_i)$. Our goal is to compute estimates $\tilde{\mathbf{x}}(t_i)$ that minimizes some cost function $\mathcal{L}\left(\boldsymbol{y}_c(t_i), \mathbf{x}(t_i), \boldsymbol{T}_f(t_0)\right)$ subject to $||\mathbf{x}(t_i) - \boldsymbol{F}\left(\mathbf{x}(t_0), \boldsymbol{u}, t_i\right)||_{\boldsymbol{P}_i} < \lambda$ where $\boldsymbol{P}_i$ is a positive definite matrix and $\lambda > 0$. In our implementation, we assume that the sensor measurement noises and random walk parameters are Gaussian distributed. Linearizing $\boldsymbol{F}$ around a nominal trajectory $\tilde{\mathbf{x}}(t)$, we derive the linearized error state integration in the form

$$\delta\mathbf{x}(t_i) = \boldsymbol{\Phi}_i\delta\mathbf{x}(t_0) + \boldsymbol{\Psi}\delta\boldsymbol{u}, \tag{13}$$

where $\boldsymbol{\Phi}_i = \frac{d\boldsymbol{F}}{d\boldsymbol{x}}$ and $\boldsymbol{\Psi} = \frac{d\boldsymbol{F}}{d\boldsymbol{u}}$. Using the Gaussian noise assumption, the optimization is solved adaptively in error state domain as a *Maximum-A-Posteriori* estimator leading to the well known EKF updates.

[9]Even a tiny drift rate in the gyros results in a slowly growing tilt error. The horizontal acceleration error is $9.8\ m/s^2$ times the tilt error in radians. Double integrating this increasing acceleration error produces a position error which grows cubically in time. Thus, while small inertial sensors can maintain accuracy of a few millimeters for one second, the drift will be hundreds of meters after just a minute or so [23].

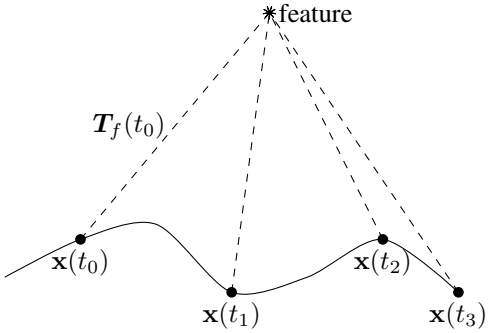

Fig. 4: Observations of Stationary Feature at Multiple Time Instants.

If the stationary feature is found to be persistent (e.g., a point tracked at the end of a corridor while navigating towards it), then we augment the feature vector along with the pose of the body frame at which it was first observed to the state $\mathbf{x}$ in order to correctly account for correlations of errors in subsequent measurements of the feature to errors in the state vector. For eg., assume that we track features $\boldsymbol{T}_{f_1}(t_1)$, $\boldsymbol{T}_{f_1}(t_2)$ (which were first observed in frames $c(t_1)$ and $c(t_2)$ respectively), then the augmented state vector $\mathbf{x}_{aug}(t)$ at some $t \geq t_1, t_2$ would be

$$\begin{aligned}\mathbf{x}_{aug}^{\top}(t) \;=\; & \left[\; \mathbf{x}^{\top}(t) \quad \boldsymbol{x}_{sb}^{\top}(t_1) \quad \boldsymbol{\Theta}_{sb}^{\top}(t_1) \quad \boldsymbol{x}_{sb}^{\top}(t_2) \quad \cdots \right. \\ & \left. \boldsymbol{\Theta}_{sb}^{\top}(t_2) \quad \boldsymbol{T}_{f_1}^{\top}(t_1) \quad \boldsymbol{T}_{f_1}^{\top}(t_2) \;\right].\end{aligned}$$

The measurements $\boldsymbol{y}_c(t)$ of features f_1 and f_2 at time instant t would be used to compute an estimate of $\mathbf{x}_{aug}$ as

$$\mathbf{x}_{aug}(t) = \arg\min_{\mathbf{x}_{aug}} \mathcal{L}\left(\boldsymbol{y}_c(t), \mathbf{x}_{aug}(t)\right),$$

subject to the dynamical constraints imposed by eq. (12).

B. Unscented Kalman Filter

To enable on–board control, a second estimator is used to estimate the full *state* of the vehicle at 200 Hz. We use an UKF, instead of an EKF because of the need to operate over a large operating envelope with significant excursions in roll and pitch angles from the hover configuration and velocities up to 3 m/s. The vehicle's state is estimated by combining the 6-DOF pose from the VIO with the IMU measurements of the phone. In the model, the state is represented by

$$\mathbf{x}_f = \begin{bmatrix}\boldsymbol{x}^{\top} & \boldsymbol{v}^{\top} & \Phi^{\top} & \boldsymbol{b}_a^{\top}\end{bmatrix}^{\top}, \tag{14}$$

where $\boldsymbol{x}$, $\boldsymbol{v}$ have been defined in Section III and the quaternion is represented by the vector

$$\Phi = \begin{bmatrix}q_0 & q_1 & q_2 & q_3\end{bmatrix}^{\top}, \tag{15}$$

and the accelerometer biases by

$$\mathbf{b}_a = \begin{bmatrix}b_{a_x} & b_{a_y} & b_{a_z}\end{bmatrix}^{\top}. \tag{16}$$

The prediction step uses the input linear acceleration and angular velocity measurements given by the phone IMU

$$\begin{aligned}\mathbf{u}_f = \begin{bmatrix}\boldsymbol{\Omega}^{\top} & \boldsymbol{a}^{\top}\end{bmatrix}^{\top},\; \mathbf{n} = \begin{bmatrix}\boldsymbol{n}_{\Omega}^{\top} & \boldsymbol{n}_a^{\top} & \boldsymbol{n}_b^{\top}\end{bmatrix}^{\top}, \\ \dot{\mathbf{x}}_f(t) = f\left(\mathbf{x}_f(t), \mathbf{u}_f(t), \mathbf{n}\right),\end{aligned} \tag{17}$$

where $\mathbf{n}$ represents the process noise that we assume to be Gaussian white noise.
Finally, the VIO pose estimates, are used to update the state estimate. We have a linear measurement model

$$\mathbf{z}_f(t) = H\mathbf{x}_f(t) + \eta, \tag{18}$$

where η is the observation noise, which is again assumed to be Gaussian white noise. The measurement model is linear since filter's update is given by the absolute position and orientation of the vehicle. The measurement delay due to the image processing is taken into account bufferizing the IMU values till a new measurement from the VIO algorithm is provided. Then, all the stored IMU measurements older than the current VIO measurement are used again in the prediction step. The separation of the VIO and UKF is useful to keep the CPU usage limited. The state size of the VIO algorithm is not constant since image features are part of the state. For this reason, running it considering a prediction and updates steps at 200 Hz is more expensive than 30 Hz. In this way, we can have similar performances and we can satisfy the control rate constraints.

V. Experimental Results

In this section we report on demonstrations at the Consumer Electronic Show (CES) 2015[10] and experiments that have been performed in the GRASP Lab [19], at The University of Pennsylvania.

The prototype shown in Fig. 1 was exhibited in CES 2015 in Las Vegas during four consecutive days with over 200 demonstrations without failures. The demonstrations were conducted against a backdrop image[11] of Las Vegas and consists of repeated vertical lines with only a few good features. In spite of this, the system performed successfully. In each demonstration, the user starts our app on the smartphone in the conventional way, which allowed the robot to take-off, hover, track pre-programmed trajectories in a $3 \times 2 \times 2$ m^3 before landing safely. As discussed before, the app incorporates real-time estimation, control and planning.

The considered working area in the GRASP Lab [19], is a volume of $5 \times 4 \times 5$ m^3. The Vicon motion capture system[12], composed of 20 T040 cameras, provides a state estimate for the quadrotor, which is considered ground truth in this context [19]. Two sets of tests were conducted with multiple repetitions — the first one considers the response of the quadrotor to step inputs. The second one is a trajectory tracking experiment to test the ability of the system to follow a trajectory in 3D space. In both experiments, the feedback control signal is obtained from the state estimation approach mentioned in Section IV at 200 Hz. Results, including controller performance and a comparison of the estimation technique with respect to the Vicon motion capture system, show the precision of the estimation of our approach during

[10] http://www.cesweb.org/

[11] http://spectrum.ieee.org/automaton/robotics/aerial-robots/a-smartphone-is-the-brain-for-this-autonomous-quadcopter

[12] http://www.vicon.com

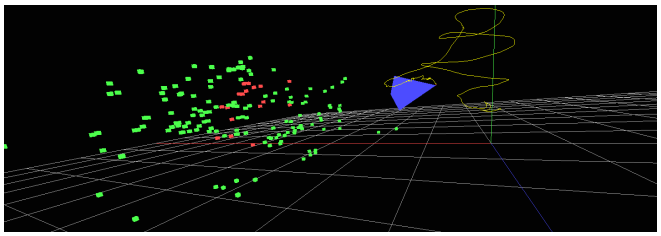

Fig. 5: Point cloud from VIO: green points were tracked in the past, red points are currently tracked, trajectory is denoted by the yellow line and camera pose by the blue tetrahedron.

flight including the ability to handle fast motions. In Fig. 5, the set of point clouds generated during a trajectory sample, is shown.

A. Step Response

The main purpose of this experiment is the evaluation of the controller performance in addition to the localization properties, which will be analyzed in Section V-B. We propose to give three step inputs of 1 m along the three axes around at time instant 3 s, 13 s, 23 s respectively along z, x, and y axis as shown in Fig. 6 and Fig. 7. The system employs an average of 1 s to reach the new desired value, which suggests a bandwidth of $0.25 - 0.5$ Hz. The velocity changes accordingly on the three axes as shown in Fig. 7. It increases in the first part of the step signal and then it decreases to the 0 value such that the vehicle is able to keep the desired value in the Cartesian space. The estimation for position and velocity components is coherent with respect to the motion capture system (blue in Fig. 6 and Fig. 7) reaching velocities of 1.3 m/s. In table I the Root Mean Square Errors (RMSE) and Standard Deviation (STD) with respect to the Vicon are reported. The average errors are 3 cm and and 0.03 m/s for the position and velocity respectively, which are sufficient for autonomous navigation. Experimental results based on trajectory tracking are provided in section V-B.

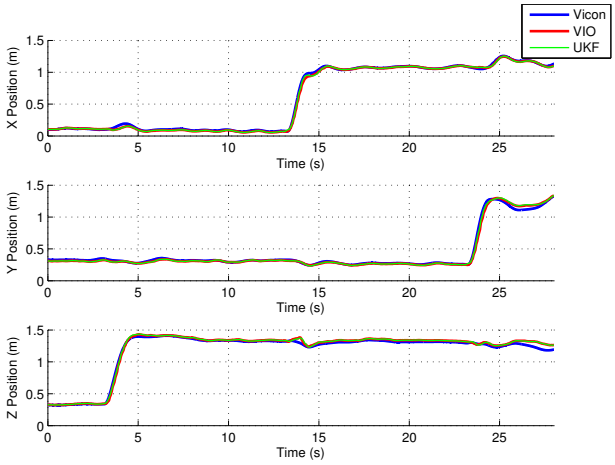

Fig. 6: Cartesian 3D position of the vehicle, with vicon (blue), VIO estimates (red) and UKF estimates (green).

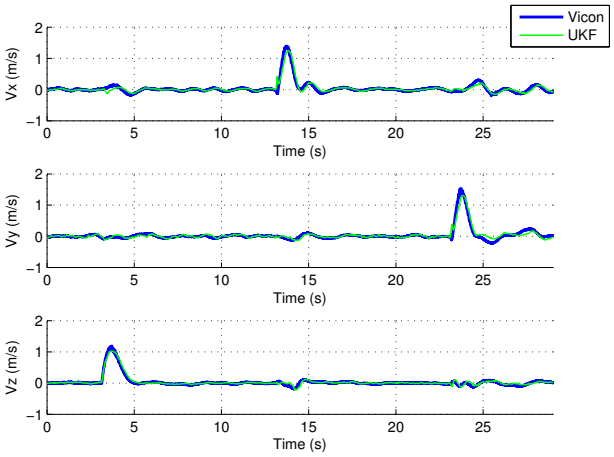

Fig. 7: Cartesian velocities of the vehicle, with the vicon (blue), VIO estimates (red) and UKF estimates (green).

B. Autonomous Navigation and Planning

In this experiment, the system is set to follow a trajectory in 3D space to test the dynamic properties of the system. The results confirm the localization properties of the presented architecture and fully validate the proposed approach for autonomous navigation in an unknown environment. The trajectory is generated according to previous work [4], [21]. Since the input $\boldsymbol{M}$ is an algebraic function of the fourth derivative of the position (snap), it is convenient to plan smooth trajectories that minimize the snap of the trajectory. This minimization problem can be formulated as a Quadratic Program (QP) [21]. Further, equality constraints can be enforced and can be determined by desired robot positions (or velocities). The trajectory that has been designed has the shape of two 8 at different values of the z coordinate. In fact, as the reader can notice from Fig. 8, the first 8 pattern is executed at the altitude of 0.7 m, while the second one at 1.2 m. The transition between the two levels is done changing the height during the 8 shape execution without hovering before the transition. During the descent as shown in Fig. 9 and Fig. 10, the vehicle stops and hovers. The same procedure is used during the take-off operation (see the z component in Fig. 9 between 3 s and 10 s where the vehicle takes off, then hovers till it moves to the first 8 shape). This is produced to stress and test entirely the proposed localization and control system. In these situations, the vehicle is still able to localize and keep the control running. In table II the values of the RMSE and STD are reported for the three axis components and for both VIO estimates and UKF estimates. We notice that the filtering technique is able to keep the same values of the VIO algorithm, while increasing the rate of the localization useful for control purposes. Along the three axis the value of the errors and STD is quite similar, but the interesting aspect to notice is that the error does not fluctuate much by decreasing the trajectory time and consequently having larger maximum velocities. This demonstrate the robustness of the estimation technique and the proposed framework for control and navigation with large

Cartesian Component	RMSE VIO estimation (m)	RMSE UKF estimation (m)	RMSE Velocity (m/s)	STD position VIO (m)	STD position UKF (m)	STD velocity (m/s)
x	0.0171	0.0156	0.0598	0.0155	0.0138	0.0591
y	0.0253	0.0237	0.0608	0.0234	0.0221	0.0608
z	0.0293	0.0287	0.0357	0.0251	0.0236	0.0355

TABLE I: Step response position and velocity RMSE and STD of the VIO and the UKF estimates compared to Vicon.

excursion in position (see Fig. 9 between 12 s and 30 s) and velocities up to 2 m/s during the 8 shape execution (see Fig. 10). We just notice a slight increase in error for the test of 2.3 m/s, but the order of magnitude is the same with respect to lower velocities. In the fastest test, the system reaches roll and pitches angles close to 30°, which causes image features to be lost and quickly re-estimated. Here as well, the system is stable and able to complete the overall task. The orientation error is evaluated according to [24] as

$$\Psi\left(\boldsymbol{R}, \boldsymbol{R}_d\right) = \frac{1}{2}\text{tr}\left(I - \boldsymbol{R}_d^T \boldsymbol{R}\right),$$

where $\boldsymbol{R}$ is the vehicle's rotation and $\boldsymbol{R}_d$ the desired one. In table III, the orientation RMSE and STD values with respect to the motion capture system are reported. They are small and of the same magnitude.

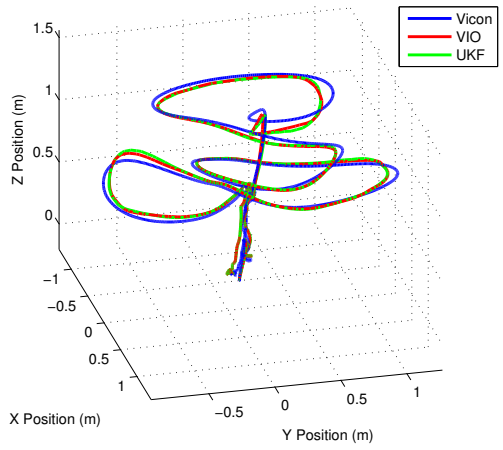

Fig. 8: Cartesian 3D position of the vehicle, with vicon (blue), VIO estimates (red) and UKF estimates (green).

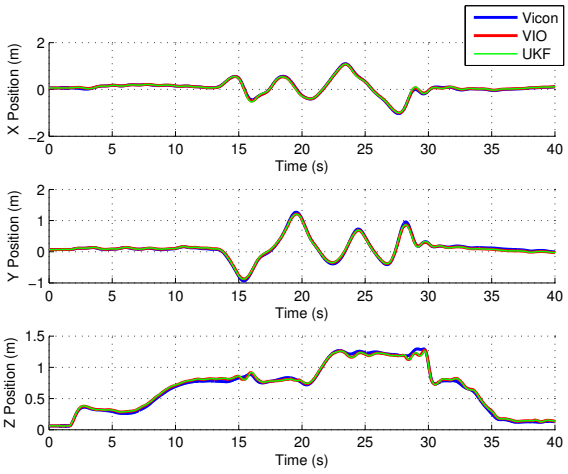

Fig. 9: Cartesian positions of the vehicle with vicon (blue), VIO estimates (red) and UKF estimates (green).

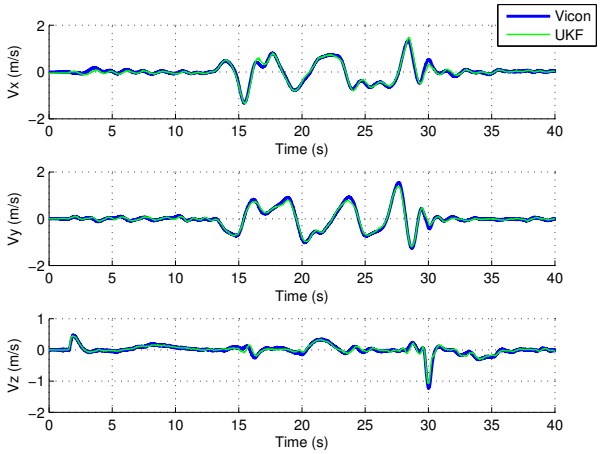

Fig. 10: Cartesian velocities of the vehicle with the vicon (blue), and UKF estimates (green).

VI. Conclusion

In this work, we have shown the hardware and software architecture with the underlying algorithms to enable the "plug and play" functionality with a consumer product (smartphone) and a quadrotor. The phone is able to solve real time control constraints, planning, mapping and estimation tasks. We demonstrated the autonomous navigation of a quadrotor platform using completely sensors onboard a smartphone such as a single camera and an IMU. With data from an external motion capture system used to measure ground truth, we demonstrated that the robot can navigate in three dimensions at average speeds of $1.5 - 2$ m/s with an average error of 3 cm in position. The use of these devices in robotics has great potential for STEM education and for personal robots. Thus, we strongly believe that smartphones will have a strong impact in real world life, for the automation of processes such as construction, package delivery and interaction between humans. Future works will investigate control and mapping strategies with multiple vehicles.

Acknowledgement

G. Loianno, Y. Mulagonkar and V. Kumar gratefully acknowledge the support of Qualcomm Research, ARL grant W911NF-08-2-0004, ONR grants N00014-07-1-0829 and N00014-09-1-1051, NSF grant IIS-1138847 and Ter-

Max. Velocity norm (m/s)	Cartesian Component	RMSE VIO estimation (m)	RMSE UKF estimation (m)	RMSE Velocity (m/s)	STD position VIO (m)	STD position UKF (m)	STD velocity (m/s)
0.5	x	0.0265	0.0243	0.0486	0.0256	0.0230	0.0481
	y	0.0277	0.0252	0.0611	0.0269	0.0243	0.0611
	z	0.0255	0.0251	0.0258	0.0230	0.0230	0.0257
0.9	x	0.0339	0.0333	0.0485	0.0205	0.0182	0.0481
	y	0.0326	0.0293	0.0623	0.0316	0.0283	0.0622
	z	0.0306	0.0307	0.0307	0.0217	0.0215	0.0307
1.8	x	0.0238	0.0244	0.0786	0.0237	0.0244	0.0781
	y	0.0336	0.0304	0.0632	0.0334	0.0302	0.0632
	z	0.0278	0.0283	0.0476	0.0263	0.0268	0.0475
2.3	x	0.0716	0.0702	0.1004	0.0562	0.0534	0.0997
	y	0.0467	0.0499	0.1183	0.0450	0.0484	0.1181
	z	0.0336	0.0348	0.0674	0.0323	0.0334	0.0662

TABLE II: Position and velocity RMSE and STD of the VIO and UKF estimates compared to Vicon at different speeds.

Orientation	Max. Velocity norm (m/s)	RMSE VIO estimation	RMSE UKF estimation	STD VIO	STD UKF
$\Psi(\boldsymbol{R}, \boldsymbol{R}_d)$	0.5	1.7888×10^{-4}	9.3579×10^{-5}	1.2384×10^{-4}	5.9579×10^{-5}
	0.9	2.5730×10^{-4}	1.5755×10^{-4}	2.009×10^{-4}	1.1557×10^{-4}
	1.8	6.9388×10^{-4}	2.1319×10^{-4}	6.3921×10^{-4}	1.8506×10^{-4}
	2.3	1.2×10^{-3}	3.2727×10^{-4}	1.1×10^{-3}	2.8741×10^{-4}

TABLE III: Orientation RMSE and STD in radians of the VIO and the UKF estimates compared to Vicon.

raSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. We would like to acknowledge the hard work of the other Qualcomm team members, A. Tyagi, A. Agarwal, P. Mendonca, T. Mathew, and Y. Chen, as well as many fruitful discussions with Prof. Stefano Soatto (UCLA).

REFERENCES

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 46–56, Sept 2012.

[2] T. Ozaslan, S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Inspection of Penstocks and Featureless Tunnel-Like Environments using Micro UAVs," in *Field and Service Robotics Conference (FSR)*, Brisbane, Australia, 2013, pp. 123–136.

[3] F. Forte, R. Naldi, and L. Marconi, "Impedance Control of an Aerial Manipulator," in *American Control Conference (ACC)*, Montreal, Canada, 2012, pp. 3839–3844.

[4] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward Image Based Visual Servoing for Aerial Grasping and Perching," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 2113–2118.

[5] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative Mapping of an Earthquake-Damaged Building via Ground and Aerial Robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[6] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[7] S. Shen, N. Michael, and V. Kumar, "Tightly-Coupled Monocular Visual-Inertial Fusion for Autonomous Flight of Rotorcraft MAVs," in *IEEE International Conference on Robotics and Automation*, Seattle, USA, 2015.

[8] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor," in *Robotics: Science and Systems (RSS)*, Berlin, Germany, 2013.

[9] K. Schmid, M. Suppa, and D. Burschka, "Towards autonomous mav exploration in cluttered indoor and outdoor environments," in *RSS 2013 Workshop on Resource-Efficient Integration of Perception, Control and Navigation for Micro Air Vehicles (MAVs)*, Berlin, Germany, 2013.

[10] S. Shen, N. Michael, and V. Kumar, "Autonomous Indoor 3D Exploration with a Micro-Aerial Vehicle," in *IEEE International Conference on Robotics and Automation*, St. Paul, USA, 2012, pp. 9–15.

[11] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[12] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, Jan 2011.

[13] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.

[14] A. Martinelli, "Visual-inertial structure from motion: Observability and resolvability," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 4235–4242.

[15] D. Wagner, T. Langlotz, and D. Schmalstieg, "Robust and Unobtrusive Marker Tracking on Mobile Phones," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, Cambridge, UK, 2008, pp. 121–124.

[16] G. Klein and D. Murray, "Parallel Tracking and Mapping on a Camera Phone," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, Orlando, USA, 2009, pp. 83–86.

[17] M. Li, B. H. Kim, and A. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *IEEE International Conference on Robotics and Automation*, May 2013, pp. 4712–4719.

[18] G. Loianno, G. Cross, C. Qu, Y. Mulgaonkar, J. A. Hesch, and V. Kumar, "Flying smartphones: Automated flight enabled by consumer electronics," *IEEE Robotics Automation Magazine*, vol. 22, no. 2, pp. 24–32, June 2015.

[19] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The Grasp Multiple Micro–UAV Test Bed," *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[20] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear Robust Tracking Control of a Quadrotor UAV on SE(3)," *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, 2013.

[21] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," in *IEEE International Conference on Robotics and Automation*, Shangai, China, 2011, pp. 2520–2525.

[22] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[23] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, 2005.

[24] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*, ser. Texts in Applied Mathematics. New York-Heidelberg-Berlin: Springer Verlag, 2004, vol. 49.