Flying Smartphones

Automated Flight Enabled by Consumer Electronics

By Giuseppe Loianno, Gareth Cross, Chao Qu, Yash Mulgaonkar, Joel A. Hesch, and Vijay Kumar

Consumer-grade technology seen in cameras and phones has led to the price–performance ratio falling dramatically over the last decade. We are seeing a similar trend in robots that leverage this technology. A recent development is the interest of companies such as Google, Apple, and Qualcomm in high-end communication devices equipped with such sensors as cameras and inertial measurement units (IMUs) and with significant computational capability. Google, for instance, is developing a customized phone equipped with conventional as well as depth cameras. This article explores the potential for the rapid integration of inexpensive consumer-grade electronics with the off-the-shelf robotics technology for automation in homes and offices. We describe how standard hardware platforms (robots, processors, and smartphones) can be integrated through simple software architecture to build autonomous quadrotors that can navigate and map unknown, indoor environments. We show how the quadrotor can be stabilized and controlled to achieve autonomous flight and the generation of three-dimensional (3-D) maps for exploring and mapping indoor buildings with application to smart homes, search and rescue, and architecture. This opens up the possibility for any consumer to take a commercially available robot platform and a smartphone and automate the process of creating a 3-D map of his/her home or office.

Digital Object Identifier 10.1109/MRA.2014.2382792
Date of publication: 11 May 2015

IMAGE LICENSED BY GRAPHIC STOCK

1070-9932/15©2015IEEE

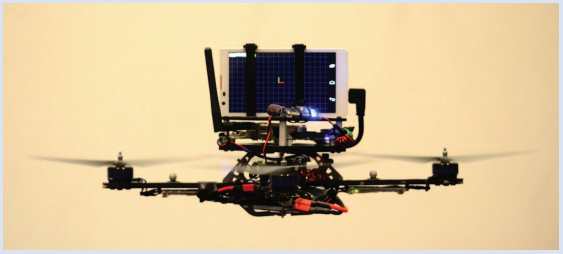
Automation of Aerial Robotics

The price–performance ratio of processors, sensors, and networking infrastructure, which has dropped significantly over the last decade, has led to new applications founded on the convergence of computation, sensing, and computing. A recent General Electric report [1] calls this convergence the *industrial Internet* and suggests that the potential macroeconomic benefit from the industrial Internet could be comparable with the economic productivity gain attributable to the Internet revolution of the late 20th century. Others call it the *Internet of Things* and predict an economic impact in the tens of trillions of dollars [2]. More than 75% of business leaders surveyed predicted a direct impact of this technology on their business [3]. This convergence also holds great promise for automation with robots, which "emphasizes efficiency, productivity, quality, and reliability, focusing on systems that operate autonomously, often in structured environments over extended periods" [4].

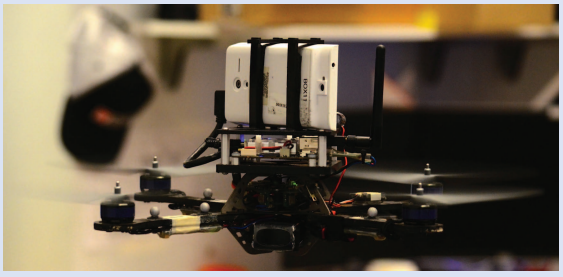
In this article, we address this confluence of technologies enabling automation in the context of aerial robotics, a field that has also seen dramatic advances over the last decade. The same drop in price–performance ratio of processors and sensors has fueled the development of micro unmanned aerial vehicles (UAVs) that are between 0.1 and 1 m in length and 0.1–2 kg in mass. These low-cost platforms are easy to manufacture in contrast to the expensive UAVs used for military applications. The number of Predators and Global Hawks is estimated to be around 1,000. Growth in the consumer electronics industry (millions or billions of components at low cost) has resulted in inexpensive hardware for sensing and computation. These advances, coupled with open-source tools for building robots like quadrotors [5], have led to innovative low-cost toys and hobby kits (e.g., diydrones.org). The real opportunity for robotics and automation is in leveraging these tools (and the convergence of computation, sensing, and communication) to develop economical (compared with the military counterparts), functional, and robust aerial robots. These can be used in such tasks as inspection [6], interaction with the environment [7], [8], search and rescue [9], construction [10], [11], and mapping of homes and offices.

Early quadrotor vehicles were primarily experimental systems, but improved design and software tools have lead to significant increases in reliability and reductions in cost. Today, quadrotors have reached the maturity of consumer-grade devices. To reiterate, this is in large part due to the decreasing price–performance ratio of sensors for autonomous navigation, including global positioning systems (GPSs), cameras, IMUs, and laser scanners [7], [12]–[16]. In this context, low-cost range sensors offer an attractive alternative to high-end laser scanners and 3-D cameras for applications such as indoor navigation and mapping, surveillance, and autonomous robotics.

Consumer-grade range sensing technology has led to many devices becoming available on the market, like the Microsoft Kinect sensor and the ASUS Xtion sensor (PrimeSense 2010; see Figure 1). The richness of the provided data and the low cost of the sensor have attracted many researchers from the fields of mapping, 3-D modeling, and reconstruction. The ASUS Xtion sensor boasts a lower weight than the first



(a)



(b)

Figure 1. The flying platform (a) back view and (b) front view.

generation of red, green, blue, and depth (RGB-D) cameras (around 70 g without the external casing). (While this specific sensor is no longer available, there are others under development that are likely to be available in the future.) It does not require external power beyond a standard universal serial bus (USB) port, and it is quite compact. Consequently, sensors in this form factor have received significant attention from the scientific community, particularly for environment mapping and monitoring applications with UAVs [16], [17].

This article addresses the use of an off-the-shelf quadrotor platform with a commercial off-the-shelf (COTS) RGB-D camera (see Figure 2). The current algorithms based on RGB-D cameras need a large platform and specific customized hardware processors that are not widely available. These algorithms generally only work without limitation on laptop or desktop computers [18], [19]. Moreover, there is still a gap between the use of complex algorithms in the research field and their use by naive humans for everyday applications. Google's Project Tango has bridged these gaps by creating a prototype [20] smartphone

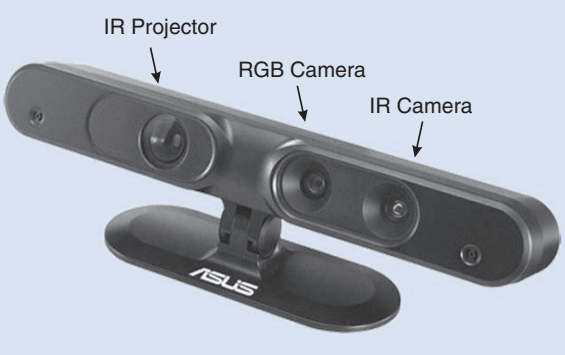


Figure 2. The ASUS Xtion Pro live sensor.

Figure 3. The Google Project Tango device.

[Google (http://www.google.com) 2014; see Figure 3]. The smartphone incorporates an enhanced RGB-D camera with a fisheye lens that has a field of view of 170° and a depth sensor able to capture a dense set of point clouds. It also incorporates customized hardware and software designed to track full 3-D motion while concurrently creating a map of the environment using visual odometry and structure from motion algorithms. These sensors allow the phone to make more than a quarter of a million 3-D measurements every second, updating its position and orientation in real time and combining the data into a single 3-D model of the surrounding space. In comparison, previous works on implementing vision-based algorithms on camera phones are based on marker tracking [21] and localization algorithms [22]. These algorithms are designed for augmented-reality applications; they employ cameras with a limited frame rate and with a small field of view. For these reasons, they are not suitable to deal with the long-term operations and large navigation coverage areas needed in robotic tasks.

In this article, a complete architecture representing a first step toward autonomous aerial robot flight with a camera phone is presented. 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 in homes by providing services for localization, state estimation, control, and mapping. In the future, end users may be able to utilize their smartphone device to autonomously control an aerial platform and to add new functionalities. The first contribution of this work is the development of a quadrotor platform equipped with the Google Project Tango [20] smartphone sensor and a small processor unit. The second contribution is the vehicle's control based on smartphone localization estimation. A nonlinear controller guarantees the exponential stability of the platform, which is able to follow trajectories in 3-D space. Finally, the fusion of the phone's pose with inertial sensor measurements allows for an increased rate of state estimation, and, thus, it enables fast motions.

System Architecture

Our platform of choice was a quadrotor due to its mechanical simplicity [5] and ease of control. Moreover, its ability to operate in confined spaces, hover at any given point in space, and perch or land on a flat surface makes it a very attractive aerial platform with tremendous potential. A description of the proposed hardware and software architecture is presented here. A schema of the proposed approach is shown in Figure 4.

Hardware Architecture

The experimental platform shown in Figure 5 is made from COTS components and is equipped with an AutoPilot board consisting of an IMU and a user-programmable ARM7 microcontroller. The main computation unit on board is an ODROID-XU (http://www.hardkernel.com) with a 1.7-GHz Samsung Exynos 5 Octa processor with 2 GB of random access memory (RAM), a 16-GB embedded multimedia controller (eMMC) module, and a 802.11n Wi-Fi transceiver. The only other addition to this setup is a forward-pointing Project Tango [20] smartphone from Google and a USB 3.0 cable for communication between the ODROID-XU and the Tango phone. The total mass of the platform is 900 g. The software framework is presented in the "Software Architecture" section.

It should be noted that our experimental setup is independent of the specifics of the employed embedded board. The processor usage is estimated to be 35% of the total available central processing unit (CPU), which suggests that a smaller and less powerful embedded processor will suffice. However, to guarantee a reliable setup and to reuse the same configuration for other robotic tasks, we choose to use the ODROID-XU

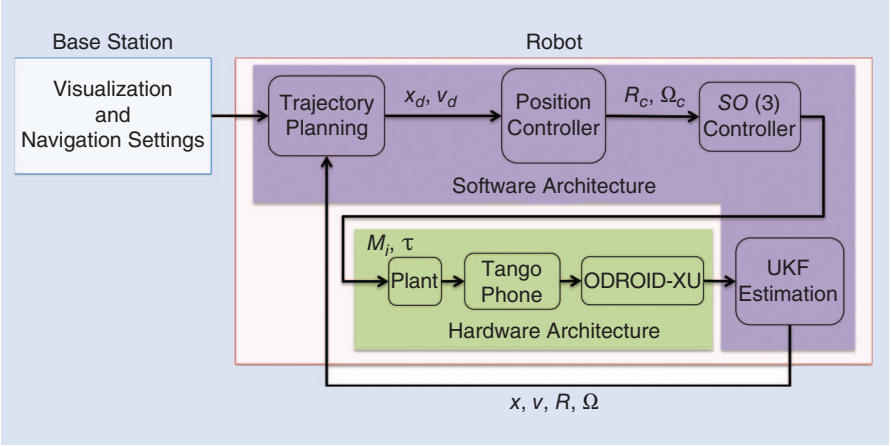


Figure 4. The system architecture for specification, planning, control, and estimation. UKF: unscented Kalman filter.

Figure 5. The computer-aided design (CAD) model for the robot platform and the Google Tango device.

board. The price is similar to other embedded platforms in the same class. Recent developments by ODROID suggest the possibility of using the less powerful ODROID-W board since the price is four times lower compared with the Odroid-XU, but the ODROID-W has been recently released on the market.

Software Architecture

The ODROID-XU performs the following tasks:

- sensor fusion between Tango's pose and the vehicle's IMU
- nonlinear position-based control.

While both tasks could conceivably have been performed on the Tango device, we used an independent processor to facilitate the ease of prototyping and to ensure a more reliable approach to state estimation and control at a fixed rate of 100 Hz.

A Java application routine is enabled on the phone for pose streaming using the user datagram protocol (UDP). The ODROID-XU runs a robot operating system (ROS)-based architecture (http://www.ros.org). The UDP packets are received by a ROS node and are subsequently converted into ordinary ROS messages. It should be pointed out that the presented strategy allows the vehicle to run all the algorithms on board. The base station is responsible only for visualization and handling user interaction.

Modeling

A quadrotor is a system made of four identical rotors and propellers located at the vertices of a square. The first and the third propellers rotate clockwise, and the second and the fourth propellers rotate counterclockwise (see Figure 6). The symbols used in this article are listed in Table 1.

Dynamic Model

Let us consider an inertial reference frame denoted by $\{\vec{e}_1; \vec{e}_2; \vec{e}_3\}$ and a body reference frame centered in the center of mass (COM) of the vehicle denoted by $\{\vec{b}_1; \vec{b}_2; \vec{b}_3\}$. The dynamic model of the vehicle can be expressed as

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

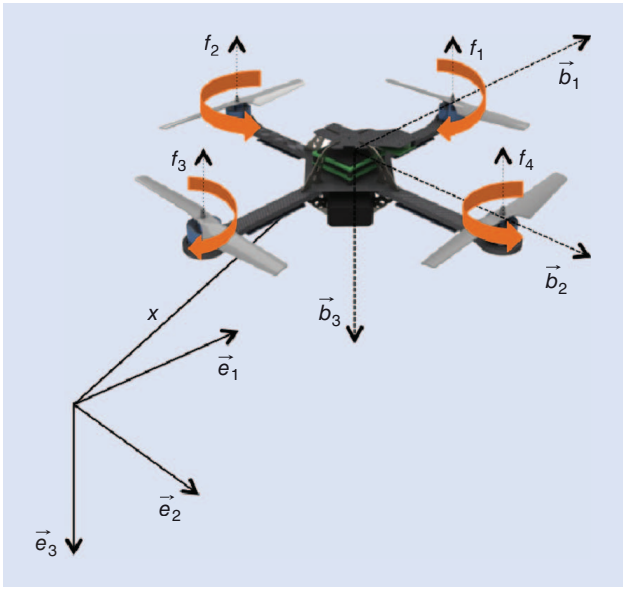


Figure 6. The quadrotor model.

Table 1. A glossary of important symbols.

$f_j \in \mathbb{R}$	Force produced by the jth propeller
$\tau \in \mathbb{R}$	Sum of forces produced by all four propellers
$\boldsymbol{M} \in \mathbb{R}^3$	Moments generated by propellers around body axes
$\boldsymbol{x} \in \mathbb{R}^3$	Position of robot's COM
$\boldsymbol{R} \in SO(3)$	Rotation matrix of the vehicle with respect to the inertial frame
$\boldsymbol{R}_c \in SO(3)$	Commanded rotation matrix
$m \in \mathbb{R}$	Mass of the vehicle
$\boldsymbol{J} \in \mathbb{R}$	Rotational inertia of robot about its COM
$\boldsymbol{\Omega} \in \mathbb{R}^3$	Angular velocity of robot in the body frame
$\boldsymbol{\Omega}_c \in \mathbb{R}^3$	Commanded angular velocity of robot in the body frame
$\boldsymbol{a} \in \mathbb{R}^3$	Linear acceleration of robot in the body frame
$g \in \mathbb{R}$	Gravitational acceleration
$d \in \mathbb{R}$	Distance of each rotor from the COM
$\boldsymbol{x}_d \in \mathbb{R}^3$	Desired position
$\dot{\boldsymbol{x}}_d \in \mathbb{R}^3$	Desired velocity
$\ddot{\boldsymbol{x}}_d \in \mathbb{R}^3$	Desired acceleration
$\boldsymbol{e}_x, \boldsymbol{e}_v \in \mathbb{R}^3$	Translational errors
$\boldsymbol{e}_R, \boldsymbol{e}_\Omega \in \mathbb{R}^3$	Attitude errors
$\mathbf{x} \in \mathbb{R}^{13}$	State estimation vector
$\mathbf{a}_b \in \mathbb{R}^3$	Accelerometer biases
$\mathbf{u} \in \mathbb{R}^6$	Estimator input

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 $\boldsymbol{J} \in \mathbb{R}^3$ is the inertia matrix with respect to the body frame. The hat symbol $\hat{\cdot}$ 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 $e_3 = [0 \;\; 0 \;\; 1]^T$.

The total moment $\boldsymbol{M} \in \mathbb{R}^3$ 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 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 COM to the center of each rotor in the $\vec{b}_1, \vec{b}_2$ plane. For nonzero values of d, (2) can be inverted. Our assumption that τ and M are the inputs of the plant is, therefore, valid.

Vehicle Control

In most previous works, a back-stepping approach to control is used because the attitude dynamics can be assumed to be faster than the dynamics governing the position and linearized controllers are used for both loops [5], [13]. In this article, because we need to model large excursions from the hover position for robustness, we use a nonlinear controller based on the work of [23] and [24].

The control inputs τ and $\boldsymbol{M}$ are chosen as

$$\begin{aligned} \boldsymbol{M} = &-k_R \boldsymbol{e}_R - k_\Omega \boldsymbol{e}_\Omega + \boldsymbol{\Omega} \times J\boldsymbol{\Omega} \\ &- J(\hat{\boldsymbol{\Omega}} \boldsymbol{R}^\top \boldsymbol{R}_c \boldsymbol{\Omega}_c - \boldsymbol{R}^\top \boldsymbol{R}_c \dot{\boldsymbol{\Omega}}_c), \end{aligned} \tag{3}$$

$$\tau = -(-k_x \boldsymbol{e}_x - k_v \boldsymbol{e}_v - mg\boldsymbol{e}_3 + m\ddot{\boldsymbol{x}}_d) \cdot \boldsymbol{R}\boldsymbol{e}_3, \tag{4}$$

where $\ddot{x}_d$ is the desired acceleration; g is the gravity acceleration; k_x, k_v, k_R, and k_Ω are positive definite terms; and the subscript $\cdot_c$ denotes a commanded value. The quantities

$$\boldsymbol{e}_R = \frac{1}{2}(\boldsymbol{R}_c^\top \boldsymbol{R} - \boldsymbol{R}^\top \boldsymbol{R}_c)^\vee, \ \boldsymbol{e}_\Omega = \boldsymbol{\Omega} - \boldsymbol{R}^\top \boldsymbol{R}_c \boldsymbol{\Omega}_c \tag{5}$$

represent the orientation and angular rate errors, respectively, whereas translation errors are represented by

$$\boldsymbol{e}_x = \boldsymbol{x} - \boldsymbol{x}_d, \ \boldsymbol{e}_v = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d. \tag{6}$$

The symbol $.^\vee$ represent the *vee* map $\mathfrak{so}(3) \to R^3$, which is the inverse of the operation denoted with the $\hat{\cdot}$ symbol. If the initial attitude error is less than 90°, the zero equilibrium of the tracking errors is exponentially stable, i.e., $[\boldsymbol{e}_x; \boldsymbol{e}_v; \boldsymbol{e}_\Omega; \boldsymbol{e}_R] \equiv [\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}]$. Furthermore, if the initial attitude error is between 90° and 180°, then the zero equilibrium of the tracking errors is almost globally exponentially attractive. See [23] for convergence and stability analysis and [24] for experimental results.

State Estimation

To enable onboard control, an unscented Kalman filter (UKF) estimator is responsible for fusing vision and inertial data. This process allows us to estimate the full pose at 100 Hz, similar to [12]. The vehicle's state is estimated by combining the six-degrees-of-freedom pose from the Project Tango [20] smartphone device and the vehicle's IMU. In the model, the state is represented by

$$\mathbf{x} = [\boldsymbol{x}^T \ \ \boldsymbol{v}^T \ \ \boldsymbol{\Phi}^T \ \ \boldsymbol{b}_a^T]^T, \tag{7}$$

where the quaternion is represented by the vector

$$\boldsymbol{\Phi} = [q_0 \ \ q_1 \ \ q_2 \ \ q_3]^T, \tag{8}$$

and the accelerometer biases by

$$\mathbf{a}_b = [a_{b_x} \ \ a_{b_y} \ \ a_{b_z}]^T. \tag{9}$$

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

$$\mathbf{u} = [\Omega^T \boldsymbol{a}^T]^T, \ \mathbf{v} = [\mathbf{v}_\Omega^T \mathbf{v}_a^T \mathbf{v}_b^T]^T, \ \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k), \tag{10}$$

where $\mathbf{v}_k$ represents the process noise the we assume to be Gaussian white noise.

Finally, the Tango pose estimates are used to update the state estimate. We have a linear measurement model

$$\mathbf{z}_k = H\mathbf{x}_k + \boldsymbol{\eta}_k, \tag{11}$$

where $\boldsymbol{\eta}_k$ is the observation noise, which is again assumed to be Gaussian white noise. The measurement model is linear since the filter's update is given by the smartphone device, which provides the absolute position and orientation of the vehicle.

Experimental Results

The experimental tests have been performed in the General Robotics, Automation Sensing, and Perception (GRASP) Lab [5] at the University of Pennsylvania. The considered working area is a volume of $5 \times 4 \times 5$ m^3, and it is composed of 20 T040 motion-capture-system cameras. The Vicon motion-capture system (http://www.vicon.com) provides a state estimate for the quadrotor, which is considered ground truth in this context. It can be run at or below 375 Hz. The standard deviation for single-marker position estimates is 50 μm, which is well beyond the accuracy required for flight. Finally, it is a robust system since the provided software assumes that the markers in the models are rigidly attached, which enables the system to maintain tracking even if all but one camera are occluded. Two sets of tests are proposed—the first considers the stabilization of the quadrotor at a desired pose, even in the presence of external disturbances, and the second tests the system's ability to follow a generated trajectory. In both experiments, the feedback control signal is obtained from the estimation filtering mentioned in the "State Estimation" section at 100 Hz. The results, including controller performance and the precision of the estimation with respect to a Vicon motion capture system, are provided.

System Stabilization

In this experiment, the system is set to hover in a defined 3-D position in space. To show the effectiveness of the proposed controller and estimation pipeline, perturbations are applied to the vehicle. The vehicle promptly returns to the set point, showing the stability of the controller presented. This behavior is shown in Figure 7, where after the taking-off phase (the first 5 s), the vehicle reaches the specified altitude and the goal position is set to $p = [0.2 \;\; 2.2 \;\; 1.2]^T$ m, while $b_{1d} = [1 \;\; 0 \;\; 0]^T$. Then, three consecutive perturbations of around 0.7 m along the z-axis, y-axis, and x-axis are noticeable (see orange circles in Figure 7). The corresponding velocity values are shown in Figure 8. After each perturbation (highlighted by the orange dashed ellipse), the vehicle controller commands changes in velocity to allow the system to return to its original position. The peak velocities are around 2 m/s, and, in each case, the controller takes less than 3 s to stabilize, suggesting a closed-loop control bandwidth of around 0.25–0.5 Hz. During each disturbance, the estimator is able to identify the pose and velocity of the system (this is noticeable just comparing the estimation with respect to the Vicon data in Figures 7–9) and reacts promptly, once released, increasing the spatial velocity to return to its hovering position. The performance is compared with the motion capture system, and root mean square error (RMSE) values are reported in Table 2. These results can be considered a first benchmark for this new device. It exhibits strong localization performances, and, thus, the COTS components are suitable for robotic tasks. The orientation error is evaluated according to [25] as

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

where $\boldsymbol{R}$ is the vehicle's rotation and $\boldsymbol{R}_d$ the desired one. As shown in Table 2, the proposed filtering technique is able to keep the errors at the same order of magnitude as the Tango device, but with the additional benefit of increased estimation rate, since the pose and the velocity are updated at 100 Hz. In Table 3, we summarize the velocity RMSE for the three Cartesian components with respect to the Vicon motion-capture system. The error is small, with an average value of 0.06 m/s.

Autonomous Navigation

In this experiment, the system is set to follow a trajectory in 3-D space. In this way, the dynamic properties of the system can be fully tested. 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 works [24], [26]. Since the input 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 using the cost functional,

$$\min \int_{t_0}^{t_f} \mu_d \left\| \frac{d^4 \boldsymbol{x}_d(t)}{dt^4} \right\|^2 dt,$$

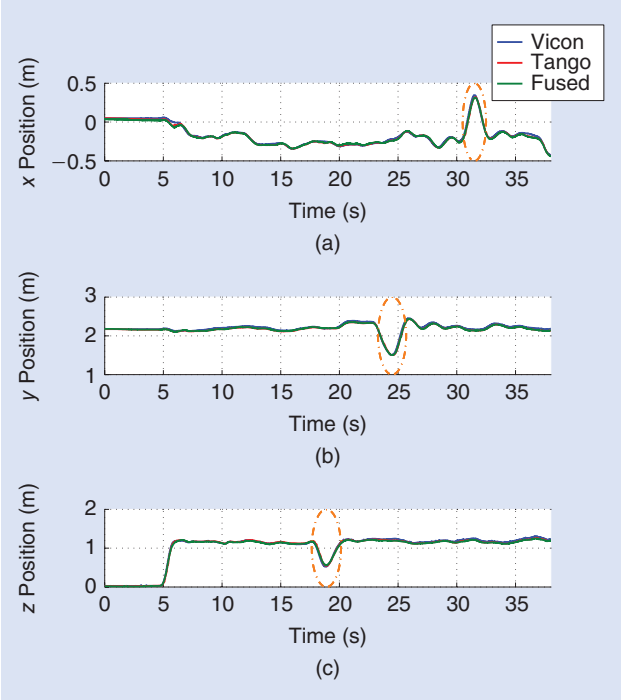


Figure 7. The Cartesian positions of the vehicle during the hovering phase (blue: Vicon, red: Tango estimation, and green: the estimator). Some jumps are noticeable in the plots (orange circles). They represent an external Cartesian perturbations made by the user to show the stability of the whole system: (a) the x Cartesian component, (b) the y Cartesian component, and (c) the z Cartesian component.

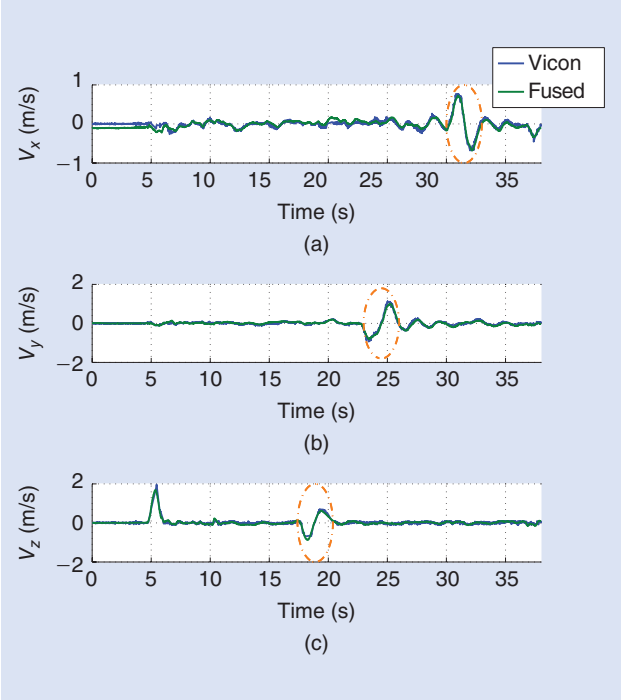


Figure 8. The Cartesian velocities of the vehicle during the hovering phase (blue: Vicon, green: the estimator). Some jumps are noticeable in the plots (orange circles). They represent the effects of external Cartesian perturbations made by the user to show the stability of the whole system: (a) the x Cartesian component, (b) the y Cartesian component, and (c) the z Cartesian component.

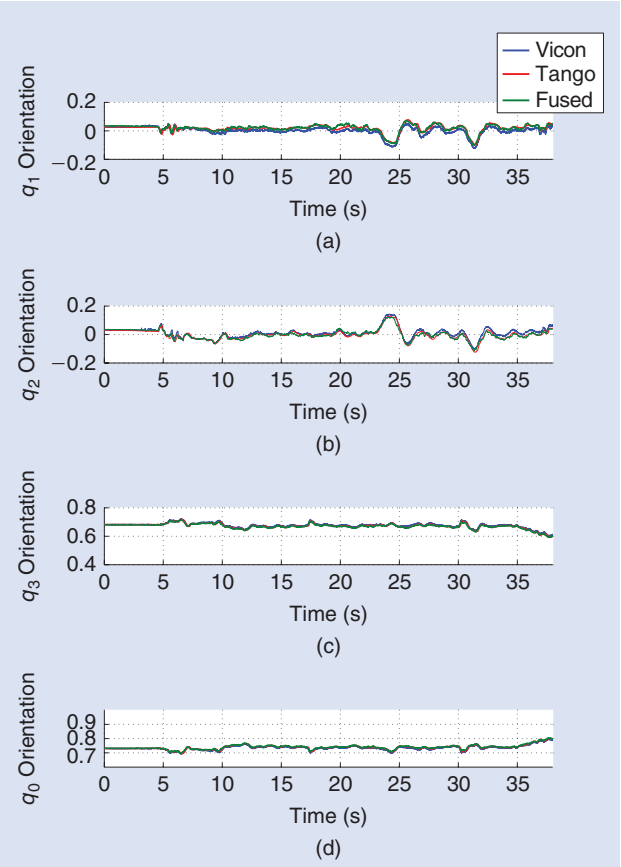


Figure 9. The orientation of the vehicle during the hovering phase (blue Vicon, red Tango estimation, and in green the estimator). Some disturbances are observable on the rotational values. Those are the effects of the vehicle recovering the initial position. (a) x Cartesian component, (b) y Cartesian component, (c) z Cartesian component, and (d) scalar.

Table 2. The position and orientation RMSE of the Tango estimation and of the estimation filtering technique compared to Vicon.

Position	Tango Estimation (m)	UKF Estimation (m)
x	0.0165	0.0195
y	0.0320	0.0296
z	0.0274	0.0299
Orientation	**Tango Estimation**	**UKF Estimation**
$\Psi(R,R_d)$	0.0627	0.091

Table 3. The velocity RMSE of the estimation filtering technique compared to Vicon.

Component	UKF Estimation (m/s)
x	0.0651
y	0.0533
z	0.0788

where μ_d is a constant value, and t_0 and t_f the inital and final time, respectively. This minimization problem can be formulated as a quadratic program [24]. Furthermore, equality constraints can be enforced and can be determined by desired robot positions (or velocities). A trajectory of 9 s, presenting an oval shape (see Figure 10), is generated passing through four points in space ($P_1 = [-1 \ \ -1.5 \ \ -1.2]^T$, $P_2 = [0 \ \ -0.5 \ \ -1.2]^T$, $P_3 = [0 \ \ 0.5 \ \ -1.2]^T$, $P_4 = [-1 \ \ -1.5 \ \ -1.2]^T$, for the x, y, and z components, respectively) according to the mentioned technique. The trajectory's sampling frequency is chosen to be the same of the feedback control signal. The velocity reaches values of 1 m/s.

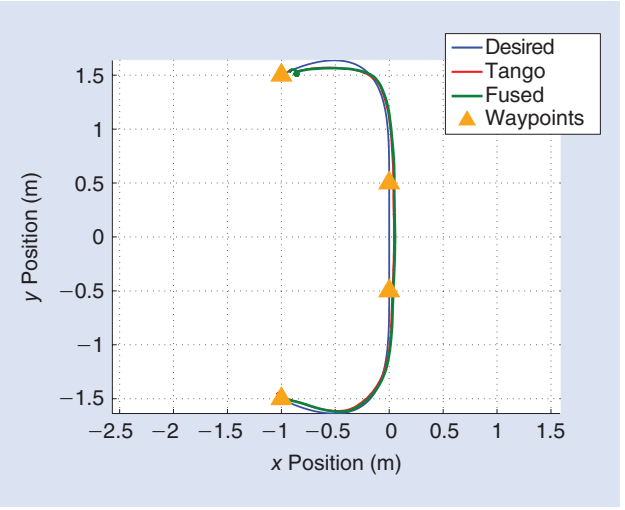


Figure 10. A view in the $x-y$ plane of the Cartesian 3-D position of the vehicle, with the desired trajectory (blue), Tango estimates (red), and UKF estimates (green).

Table 4. The position RMSE of the Tango estimation and of the estimation filtering technique with respect to the desired values.

Position	Tango Estimation (m)	UKF Estimation (m)
x	0.0593	0.0565
y	0.0719	0.0657
z	0.0122	0.0125

Table 5. The velocity RMSE of the estimation filtering technique with respect to the desired values.

Component	UKF Estimation (m/s)
x	0.0626
y	0.0833
z	0.0214

In Tables 4 and 5, the RMSE values of the Cartesian position and velocity (shown in Figures 11 and 12, respectively), with respect to the planned trajectory, are reported. The error is relatively small, with maximum values of around 6 cm, and similar to the error obtained during the stabilization task. This confirms the feasibility of using this new device and the presented approach for real robotic tasks. In

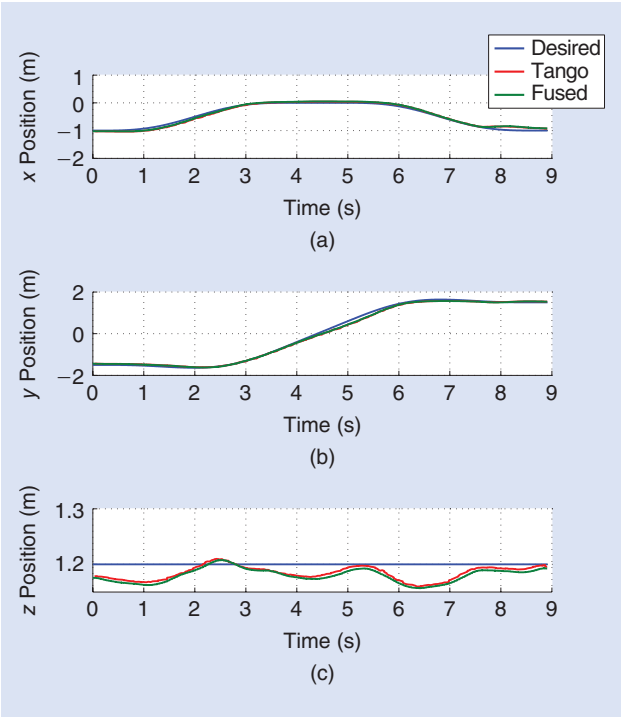


Figure 11. The Cartesian positions of the vehicle with the desired trajectory (blue), Tango estimates (red), and UKF estimates (green). (a) *x* position Cartesian component, (b) *y* position Cartesian component, and (c) *z* position Cartesian component.

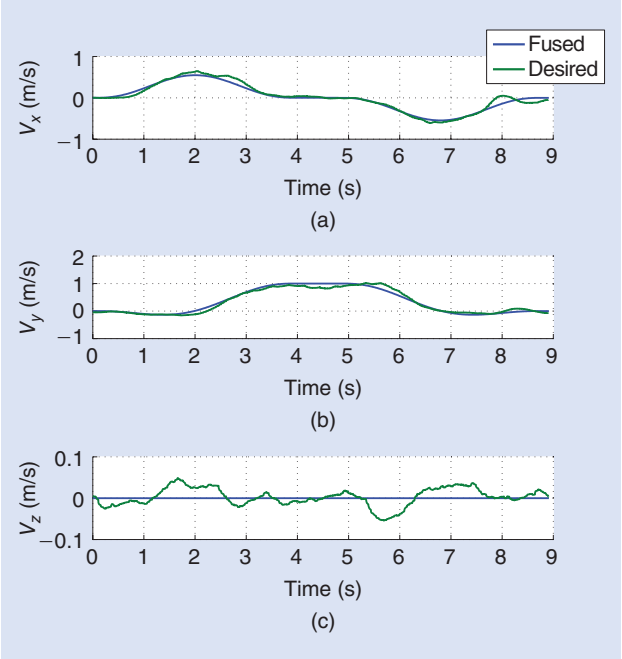


Figure 12. The Cartesian velocities of the vehicle with the desired trajectory (blue) and UKF estimates (green). (a) *x* velocity Cartesian component, (b) *y* velocity Cartesian component, and (c) *z* velocity Cartesian component.

the multimedia content available at https://www.youtube.com/watch?v=WuPZeD2J3pw, the extensive experiments are presented, with different trajectory profiles and maximum velocities reaching values up to 1.5 m/s.

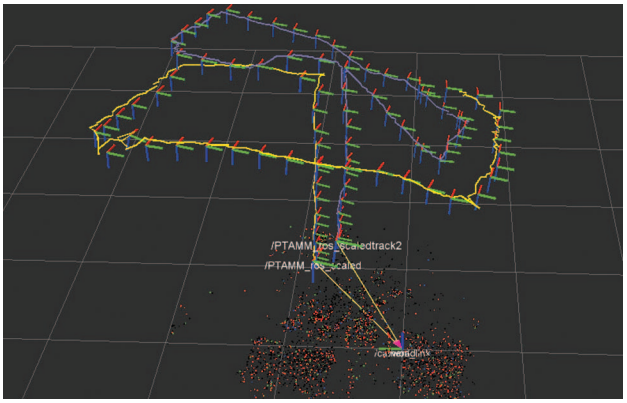


Figure 13. A representative map created by two vehicles using our RGB-D framework.

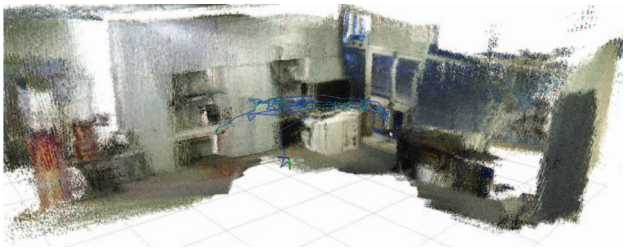


Figure 14. A representative dense map created using our RGB-D framework suggests that this vision is likely to become a reality.

Conclusions

In this article, we demonstrate the hardware and software architecture with the underlying algorithms to enable the plug-and-play functionality with a smartphone and quadrotor. In addition, we demonstrated the autonomous navigation of a quadrotor platform. With data from an external motion-capture system to measure ground truth and desired trajectory values, we show that the robot can navigate in 3-D at speeds of 1–1.5 m/s with an average error of 4 cm in position. Furthermore, we have shown that the system is robust and can recover from external disturbances of 0.7 m. Both these facts show that the system is suitable for reliable indoor navigation. A simple user interface, possibly from another handheld platform, can be used to guide the robot to designated points in a map or to generate a 3-D map. This opens up the vision of a flying robot for every human user (the number of cell phones in the world is already comparable with the world population), and the potential of using aerial robots to generate 3-D maps of buildings, localize sensors in smart buildings, take pictures from different viewpoints, and assist humans in search and rescue applications. Thus, we strongly believe that smartphones will have a strong impact in real life, in terms of advances in the automation of processes such as construction, package delivery, and interaction between humans. The interest of companies such as Google, Apple, and Qualcomm in smartphones equipped with such sensors as 3-D cameras and IMUs suggests that this vision is likely to become a reality.

While the prototype described in this article requires the use of an additional processor to facilitate integration, there is no reason why we should not be able to use the onboard processor on the smartphone. In the near future, this would allow any user the

possibility to use and control any aerial robotic platform with his or her own smartphone. With a robust and accurate control system for autonomous flight, questions of mapping and obstacle avoidance at moderate to high speeds (2–4 m/s) are realistic avenues for future research. Finally, we wish to use multiple devices to enable cooperative localization and mapping of the environment, as already developed for RGB-D cameras (see Figures 13 and 14).

Acknowledgments

This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program, a Semiconductor Research Corporation program sponsored by MARCO and the U.S. Defense Advanced Research Projects Agency; Air Force Office of Scientific Research grant FA9550-10-1-0567; Army Research Laboratory grants W911NF-08-2-0004 and W911NF-10-2-0016; Office of Naval Research grants N00014-07-1-0829, N00014-14-1-0510, N00014-09-1-1051, N00014-11-1-0725, and N00014-09-1-103; National Science Foundation grants PFI-1113830, IIS-1138847, and IIS-1328805; and the UPS Foundation.

References

[1] P. C. Evans and M. Annunziata. (2012). Industrial internet: Pushing the boundaries of minds and machines, GE white paper. McKinsey Global Inst., Washington, D.C., pp. 1–37. [Online]. Available: http://www.ge.com/docs/chapters/Industrial_Internet.pdf

[2] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs. (2013). Disruptive technologies: Advances that will transform life, business, and the global economy. McKinsey Global Inst., Washington, D.C., pp. 1–176. [Online]. Available: http://www.mckinsey.com/insights/business_technology/disruptive_technologies

[3] Economist. (2013). The Internet of Things business index, a report from the economist intelligence unit. [Online]. Available: http://www.economistinsights.com/analysis/internet-things-business-index

[4] E. Guglielmelli and K. Goldberg, "What is automation?" *IEEE Robot. Automat. Mag.*, vol. 21, no. 1, pp. 94–96, 2014.

[5] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro–UAV test bed," *IEEE Robot. Automat. Mag.*, vol. 17, no. 3, pp. 56–65, 2010.

[6] T. Ozaslan, S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Inspection of penstocks and featureless tunnel-like environments using micro UAVs," in *Proc. Field Service Robotics Conf.*, Brisbane, Australia, 2013, pp. 123–136.

[7] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *Proc. IEEE Int. Conf. Robotics Automation*, Hong Kong, China, 2014, pp. 2113–2118.

[8] F. Forte, R. Naldi, and L. Marconi, "Impedance control of an aerial manipulator," in *Proc. American Control Conf.*, Montreal, QC, Canada, 2012, pp. 3839–3844.

[9] 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," *J. Field Robot.*, vol. 29, no. 5, pp. 832–841, 2012.

[10] Aerial cooperative assembly system. (2014, June). EU collaborative project ICT-287617, "ARCAS." [Online]. Available: http://www.arcas-project.eu/

[11] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Auton. Robots*, vol. 33, no. 3, pp. 323–336, 2012.

[12] 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.

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

[14] V. Lippiello, G. Loianno, and B. Siciliano, "MAV indoor navigation based on a closed-form solution for absolute scale velocity estimation using optical flow and inertial data," in *Proc. IEEE Int. Conf. Decision Control European Control Conf.*, Orlando, FL, 2011, pp. 3566–3571.

[15] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in *Proc. IEEE Int. Conf. Robotics Automation*, Hong Kong, China, 2014, pp. 4974–4981.

[16] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3D exploration with a micro-aerial vehicle," in *Proc. IEEE Int. Conf. Robotics Automation*, St. Paul, MN, 2012, pp. 9–15.

[17] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Proc. Int. Symp. Robotics Research*, Flagstaff, AZ, 2011.

[18] G. Loianno, V. Lippiello, and B. Siciliano, "Fast localization and 3D mapping using an RGB-D sensor," in *Proc. Int. Conf. Advanced Robotics*, Montevideo, Uruguay, 2013, pp. 1–6.

[19] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGBD cameras," in *Proc. IEEE Int. Conf. Intelligent Robot Systems*, Tokyo, Japan, 2013, pp. 2100–2106.

[20] Project Tango. (2014, June). Google Project Tango. [Online]. Available: https://www.google.com/atap/projecttango

[21] D. Wagner, T. Langlotz, and D. Schmalstieg, "Robust and unobtrusive marker tracking on mobile phones," in *Proc. Int. Symp. Mixed Augmented Reality*, Cambridge, U.K., 2008, pp. 121–124.

[22] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *Proc. Int. Symp. Mixed Augmented Reality*, Orlando, FL, 2009, pp. 83–86.

[23] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE(3)," *Asian J. Control*, vol. 15, no. 2, pp. 391–408, 2013.

[24] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robotics Automation*, Shangai, China, 2011, pp. 2520–2525.

[25] F. Bullo and A. Lewis, *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. New York: Springer-Verlag, 2005.

[26] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, "Toward autonomous avian-inspired grasping for micro aerial vehicles," *Bioinspiration Biomimetics*, vol. 9, no. 2, p. 025010, 2014.

Giuseppe Loianno, University of Pennsylvania, Philadelphia, USA. E-mail: loiannog@seas.upenn.edu.

Gareth Cross, University of Pennsylvania, Philadelphia, USA. E-mail: gcross@seas.upenn.edu.

Chao Qu, University of Pennsylvania, Philadelphia, USA. E-mail: quchao@seas.upenn.edu.

Yash Mulgaonkar, University of Pennsylvania, Philadelphia, USA. E-mail: yashm@seas.upenn.edu.

Joel A. Hesch, Google Inc., Mountain View, California, USA. E-mail: joelhesch@google.com.

Vijay Kumar, University of Pennsylvania, Philadelphia, USA. E-mail: kumar@cis.upenn.edu.