Human Gaze-Driven Spatial Tasking of an Autonomous MAV

Liangzhe Yuan, Christopher Reardon, Garrett Warnell, and Giuseppe Loianno

***Abstract*—In this letter, we address the problem of providing human-assisted quadrotor navigation using a set of eye tracking glasses. The advent of these devices (i.e., eye tracking glasses, virtual reality tools, etc.) provides the opportunity to create new, non-invasive forms of interaction between humans and robots. We show how a set of glasses equipped with gaze tracker, a camera, and an inertial measurement unit (IMU) can be used to estimate the relative position of the human with respect to a quadrotor, and decouple the gaze direction from the head orientation, which allows the human to *spatially task* (i.e., send new 3-D navigation waypoints to) the robot in an uninstrumented environment. We decouple the gaze direction from head motion by tracking the human's head orientation using a combination of camera and IMU data. In order to detect the flying robot, we train and use a deep neural network. We experimentally evaluate the proposed approach, and show that our pipeline has the potential to enable gaze-driven autonomy for spatial tasking. The proposed approach can be employed in multiple scenarios including inspection and first response, as well as by people with disabilities that affect their mobility.**

I. Introduction

MULTI-ROTOR Micro Aerial Vehicles (MAVs) such as quadrotors are very popular platforms due to their size, cost, ability to hover in place, and navigate complex 3D environments, all while providing diverse payload options. They can be employed to help humans accomplish many useful tasks such as exploration [1], inspection [2], mapping [3], interaction with the environment [4], and search and rescue [5]. One crucial trait that MAV systems that aim to achieve this vision require is *autonomy*, e.g., the ability to operate without external infrastructure such as GPS or motion capture systems that are typically absent from the environments where these systems may be of the greatest use. While building MAVs that exhibit more and more autonomy has indeed been the subject of a great deal of research over the past decade, autonomy alone is not sufficient for these systems to be useful. In addition, MAVs must also be able to interact with their human counterparts in effective ways.

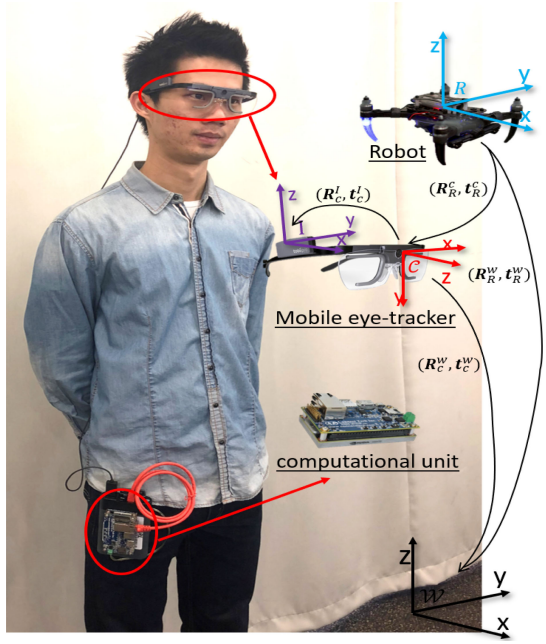

Fig. 1. A user wearing the Tobii Pro Glasses 2, where the camera frame (red) and the IMU frame (purple) are both attached to the glasses. The black reference frame denotes the world fixed frame with z-axis is aligned to the gravity direction, and the blue frame denotes the robot body frame.

The human-MAV interaction we are particularly concerned with in this letter is that of *spatial tasking*, i.e., a human tasking the MAV to navigate to or perform some other task with respect to a particular spatial location (e.g. interact with the environment or inspect an area). In current commercial MAV systems, spatial tasking is accomplished either by the human teleoperating the MAV, or by the human manually specifying a GPS coordinate as a goal. Generally, neither of these spatial tasking paradigms is desirable. In the former case, the burden of teleoperation means that the human is not able to perform other tasks, thus lowering the efficacy of the team as a whole. The latter case is undesirable because it relies on GPS availability and accuracy, and it is often difficult for a human to translate a desired location within their spatial field of view to a GPS coordinate in real time. Taking advantage of the recent availability of accurate, wireless, streaming, and on-head eye-tracking glasses (e.g., we use the Tobii Pro Glasses 2 [6] depicted in Fig. 1), we believe that the use of human gaze information can be one way to increase

Manuscript received September 10, 2018; accepted January 9, 2019. Date of publication January 25, 2019; date of current version February 15, 2019. This letter was recommended for publication by Associate Editor F. Ruggiero and Editor J. Roberts upon evaluation of the reviewers' comments. This work was supported by Qualcomm Research and the ARL under Grant DCIST CRA W911NF-17-2-0181. *(Corresponding author: Liangzhe Yuan.)*

L. Yuan is with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: lzyuan@seas.upenn.edu).

C. Reardon and G. Warnell are with the U.S. Army Research Laboratory, Adelphi, MD 20783 USA (e-mail: christopher.m.reardon3.civ@mail.mil; garrett.a.warnell.civ@mail.mil).

G. Loianno is with the New York University, Tandon School of Engineering, Brooklyn, NY 11201 USA (e-mail: loiannog@nyu.edu).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2019.2895419

the effectiveness of human-MAV interactions in human robot collaborative tasks.

Of inspiration to us in this context is the relative ease with which human-human teams are able to accomplish this same task. That is, two humans located in the same environment are typically able to communicate quickly and efficiently with one another regarding specific spatial locations using several means of communication. Of particular to interest to us here is the information provided by human *gaze*, i.e., the current pointing direction of the eyes. Studies from the psychology literature (e.g., [7]–[9]) have suggested that there is a strong link between a human's ability to perceive where a human partner is looking and their ability to infer that partner's intention or goal toward a particular object. The gaze information, despite pupils' limited range of motion, can complement, augment and possibly help to predict human speech [10] and gesture actions [11] or to disambiguate uncertain types of interactions [12], [13]. These aspects suggest that gaze can contribute to, refine, and speed up human-robot interaction tasks.

In this work, we take a first step toward enabling meaningful human gaze processing in human-MAV teams for spatial tasking. We consider a situation in which a human equipped with an on-head device consisting of a gaze tracker, a camera, and an inertial measurement unit (IMU) is co-located with a MAV teammate in an otherwise uninstrumented environment. Specifically, we propose solutions to several fundamental sensor-processing issues that must be solved in order to translate gaze information from the format of the head sensor to a format that is of use to the MAV. Additionally, we demonstrate a prototype system in which a human can control a flying robot using their eye gaze. Our contributions are:

- We provide a technique for decoupling gaze direction from head attitude.
- We present a novel object-detection-based approach to MAV localization in the human sensor frame.

These two techniques give us the ability to demonstrate, for the first time, that an autonomous flying robot can be tasked, in an uninstrumented environment, using gaze even when the human is not in the robot's field of view. Importantly, our prototype system is lightweight and does not require a ground station since the user carries only a small Jetson TX2 module and the MAV navigates using on-board Visual Inertial Odometry (VIO).

The paper is organized as follows. Section II introduces the previous works in the field. Section III presents a system overview of the main components. Section IV, presents the proposed approach used to decouple head orientation estimation from the human gaze, and how a neural network is used to detect the drone (denoted in the following as MAV, aerial robot/vehicle or simply robot) during flight. In Section V, we evaluate our localization approach and the sensor characteristics with respect to a motion capture system. Finally, Section VI concludes the work and proposes future development of scenarios that can be enabled by the proposed pipeline.

II. Related Work

A large part of the human-robot interaction (HRI) literature focuses on attentional mechanisms enabled by robots with traits similar to humans, especially considering gaze. An excellent survey on this topic is provided in [14]. Our work addresses a different problem involving a sensor-processing methodology, with the goal of enabling control of an autonomous MAV using human gaze. The use of aerial platforms closely collaborating with human operators alongside humans in uninstrumented, fully 3D environments presents a wide range of research challenges in HRI that remain unsolved.

Previous approaches using human interaction to direct MAVs in 3D space have largely focused on gestures. Gesture modalities can include a dictionary of spatial gestures presented with colored gloves [15], using gesture "metaphors" including pointing, mimicking joystick controls, mimicking manipulation of the MAV as if it were within the human's grasp [16], and falconeering-inspired gestures [17]. Augmented reality (AR) or virtual reality (VR) as an interaction modality is of increasing interest as well, e.g., using head position detected via the AR head-mounted-device (AR-HMD) combined with hand gestures detected by the AR-HMD camera [18]. The coupling between VR and drone for control was analyzed in [19]. These previous efforts in AR and VR for interaction employed instrumented environments (e.g., motion capture or fiducials on agents and objects), with the exception of [20], which used AR to interact with a single ground robot in an uninstrumented environment.

There is also earlier work exploring using gaze for teleoperating MAVs. In these works, though, gaze is always associated with the head motion (and not eye pointing direction), which is not always a valid assumption. It has been shown [21] that the head orientation is not a good indicator of the gaze direction, mainly because people orient to object with saccades [22], especially when interacting with lateral objects. In [23] the authors used gaze gestures to allow a user to teleoperate a drone via a screen and an eye tracking interface (e.g., for translation - up, down, left, right - and rotation). In [24], the authors took an interesting approach to determine which two of the four degrees of freedom when teleoperating a MAV could be best controlled via x-y gaze movement, leaving the other two to be controlled via keyboard. They found that rotation and speed controlled by gaze and translation and altitude by keyboard were the most reliable combination. Similar in spirit to this work, gaze has recently been studied as a method by which passengers might task autonomous vehicles for problems such as en-route destination changing [25], [26]. Three works closely related to our efforts are [27], [28], and [29]. In [27] the authors use face orientation to calculate a projected trajectory for a MAV to fly and facial expression to select between trajectory types, whereas [28] employs a face-angle distance plus gesture to direct MAV motion. In [29], the authors use vision to add or remove robots from teams and gestures to execute team behavior. In all these works, the gaze is again associated with the head orientation. The decoupling between human gaze and head orientation can be useful to direct a MAV in 3D space in a more refined way.

III. System Overview

In this section, we discuss the system hardware and algorithm components of our pipeline. A detailed diagram describing our

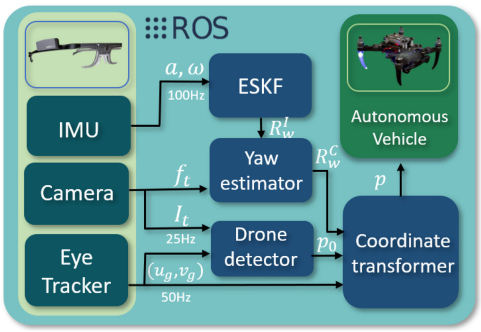

Fig. 2. The proposed architecture for gaze-driven spatial tasking, with glasses sensors (light yellow box), ROS modules (blue boxes) and autonomous vehicle (green box). The IMU and camera provide inertial measurements $(\mathbf{a}, \boldsymbol{\omega})$, and visual features f_t, for state estimation. The image I_t is used by the drone detector to estimate the human/robot relative position.

pipeline is shown in Fig. 2. The framework has been developed in ROS.[1]

A. Hardware Setup

Our novel system is based on the Tobii Pro Glasses 2, a portable head mounted eye tracker, and a computation unit, which is connected directly to the glasses. On the Tobii Glasses, an outward-looking scene camera provides 25 Hz HD images and a built-in MEMS sensor provides linear acceleration and angular velocity at ∼100 Hz. More importantly, the glasses provide a gaze location estimation $(u_g, v_g)^\top$ on the image plane at 50 Hz. For the computation unit, we use an NVIDIA Jetson TX2 as a portable platform which provides both CPU and GPU for running the estimation and detection algorithm online. For the MAV, we use a 250 g platform from our previous work [30].

B. Algorithm Pipeline

An Error-State Kalman Filter (ESKF) at 100 Hz is used to process IMU data and estimate the attitude of glasses. It is well-known that the full attitude cannot be recovered; the yaw angle with respect to the reference frame will drift over time due to unobservability. Therefore, given the relative pose from the ESKF with two known orientation angles, we use a vision-based, five point linear algorithm to compensate for this yaw drift. To localize the drone in the glasses camera's field of view, we train and deploy a deep neural network. The detection algorithm can process a single frame in 330 ms on TX2, but we only perform detection during gaze fixations (here, when the user's gaze is fixed for more than 200 ms on the image plane). Given a fixation, we use the drone detector, and the system selects the drone if the fixation location and drone location coincide. Once the drone has been selected, we then estimate its position relative to the human based on the detector's bounding box and the known physical size of the agent. This position estimate, combined with onboard drone odometry, is then used to estimate the human's position in the world reference frame. Finally, the user can move his/her gaze somewhere else and virtually select a navigation waypoint for the MAV.

[1]www.ros.org

IV. Methodology

In this section, we describe our approach to estimating the head orientation using the glasses IMU and camera, which is employed to compensate for yaw drift and detect the drone.

A. Notation

As shown in Fig. 1, we define the camera frame denoted with $\mathcal{C}$, the IMU frame denoted with $\mathcal{I}$, the robot frame denoted with $\mathcal{R}$, and a world frame denoted with $\mathcal{W}$. The gaze is always defined in the $\mathcal{C}$ frame, whereas accelerometer and gyro data in the frame $\mathcal{I}$. Without loss of generality, to simplify the notation, in the following we assume that the IMU and camera frames are coincident. Their relative pose is obtained with a calibration procedure detailed in [31]. The notation $R_{\mathcal{B}}^{\mathcal{A}}$ or $\mathbf{q}_{\mathcal{B}}^{\mathcal{A}}$ defines a rotation (expressed as matrix R or quaternion $\mathbf{q}$), which converts a point from the frame $\mathcal{B}$ to the frame $\mathcal{A}$.

B. Error-State Kalman Filter

The head attitude is estimated using an Error-State Kalman Filter (ESKF). The state is represented by

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_{\mathcal{I}}^{\mathcal{W}\top} & \mathbf{b}_\omega^\top \end{bmatrix}^\top, \tag{1}$$

where $\mathbf{q}_{\mathcal{I}}^{\mathcal{W}}$ is the orientation of the frame $\mathcal{I}$ with respect to the $\mathcal{W}$ frame expressed in quaternion form and $\mathbf{b}_\omega$ is the gyro bias. Given the orientation $\mathbf{q}_{\mathcal{I}}^{\mathcal{W}}$, a rotation increment $\delta\mathbf{q}$ from the current orientation gives a new orientation $\bar{\mathbf{q}}$ as

$$\bar{\mathbf{q}} = \mathbf{q}_{\mathcal{I}}^{\mathcal{W}} \otimes \delta\mathbf{q} = \mathbf{q}_{\mathcal{I}}^{\mathcal{W}} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix}, \tag{2}$$

where $\delta\theta$ is the angle-axis difference between the estimated attitude and the true one as specified in [32], and the operator $\otimes$ indicates the quaternion multiplication. It follows that the IMU error-state is defined as

$$\delta\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\theta}^\top & \delta\mathbf{b}_\omega^\top \end{bmatrix}^\top. \tag{3}$$

The error-state process model, as in [32], can be written as

$$\dot{\delta\boldsymbol{\theta}} = -[\boldsymbol{\omega}_m - \mathbf{b}_\omega]_\times \delta\boldsymbol{\theta} - \delta\mathbf{b}_\omega - \omega_n, \quad \dot{\delta\mathbf{b}}_\omega = \eta_{\mathbf{b}_\omega}, \tag{4}$$

where the $[]_\times$ operator converts a vector into its corresponding skew symmetric matrix. The variable ω_n is the process noise assumed to be Gaussian white noise, $\eta_{\mathbf{b}_\omega}$ is Gaussian white noise as well since the gyro bias is modeled as random walk process. The measured $\boldsymbol{\omega}_m$ angular rate is modeled as

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_\omega + \omega_n, \tag{5}$$

where $\boldsymbol{\omega}$ is the true angular rate. The process model is discretized using a first order Euler integration scheme.

In our pipeline, the measurement update is given by the gravity acceleration expressed in the frame $\mathcal{I}$ as

$$\mathbf{z} = \mathbf{q}_{\mathcal{W}}^{\mathcal{I}} \otimes \begin{bmatrix} 0 & 0 & 0 & g \end{bmatrix}^\top \otimes \mathbf{q}_{\mathcal{W}}^{\mathcal{I}} + \eta_z, \tag{6}$$

with g the gravity acceleration and η_z is the measurement noise, which is assumed to be Gaussian. In the update step, we use the extended Kalman filter equations and compute the Jacobian of eq. (6) with respect to the error state in eq. (3).

C. Vision-Based Yaw Estimation

As discussed in the last section, the ESKF is able to provide an estimation of the relative orientation between the $\mathcal{I}$ and $\mathcal{W}$ frames. However, using only an inertial sensor, the approach would suffer from yaw drift. To mitigate the effect of yaw unobservability, we use another measurement to correct the drift. The front facing camera, as shown in Fig. 1, allows us to form a camera-IMU system and fully localize the head orientation. The yaw gets estimated using a 5-points algorithm similar to [33]. Given two image frames I_1 and I_2, we first extract FAST features [34] in the image I_1, and then track them in the second frame using the KLT tracker [35]. We assume that the camera is calibrated, which allows us to compensate for the distortion of tracked features and use normalized image coordinates. We denote the set of coordinates in image frames I_1 and I_2 with $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ respectively. We decompose the rotation matrix $R_{I_1}^{I_2}$ into Euler angle rotations using the convention Z-Y-X

$$R_{I_1}^{I_2} = R_z R_y R_x, \tag{7}$$

where R_z, R_y, and R_x are the rotation matrix along each Cartesian axis, respectively. R_y and R_x can be estimated from the ESKF. Let $\hat{p}_1$ denote the undistorted points p_1 rotated by R_y and R_x. Then the epipolar constraint between the two frames I_1 and I_2 can be expressed as

$$\boldsymbol{p}_2^\top [\mathbf{t}]_\times R_z (R_y R_x \boldsymbol{p}_1) = \boldsymbol{p}_2^\top E \hat{\boldsymbol{p}}_1 = 0, \tag{8}$$

where the essential matrix $E = [\mathbf{t}]_\times R_z$ in this case, has a simpler form. Noticing its structure

$$E_{3,3} = 0,\ E_{1,2} = -E_{2,1},\ E_{1,1} = E_{2,2}, \tag{9}$$

we can rearrange eq. (8) with only 6 entries of E as

$$\mathbf{a}\boldsymbol{e} = 0, \tag{10}$$

with

$$\hat{\boldsymbol{p}}_1 = [x_1, y_1, z_1],\ \boldsymbol{p}_2 = [x_2, y_2, z_2] \tag{11}$$

$$\mathbf{a} = \begin{bmatrix} x_1x_2 + y_1y_2 & x_1y_2 + y_1x_2 & x_1z_2 & y_1z_2 & z_1x_2 & z_1y_2 \end{bmatrix} \tag{12}$$

$$\boldsymbol{e} = \begin{bmatrix} E_{1,1} & E_{1,2} & E_{1,3} & E_{2,3} & E_{3,1} & E_{3,2} \end{bmatrix}^\top. \tag{13}$$

Selecting 5 points, we obtain a linear system in the form

$$A\boldsymbol{e} = 0, \tag{14}$$

where we can find E looking to the A null space. To guarantee that the matrix belongs to the essential matrix space, we do a singular value decomposition of E obtaining E' as

$$E' = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top. \tag{15}$$

Finally, we decompose E' to recover rotation R and translation $\boldsymbol{t}$. Instead of projecting all points into 3D with all four solutions, we compare our yaw angle estimate γ with the prior from IMU and select the closest solution, which is faster from a computational point of view. Then, γ can be calculated as

$$\gamma = atan2(-E'_{1,1}, E'_{2,1}). \tag{16}$$

To discard incorrect feature matchings, we use a 2-points RANSAC outlier rejection scheme. Given angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ between frames from IMU, we obtain a rough estimate of the rotation between I_1 and I_2 using a zero order integration scheme via the exponential map [36]

$$R_{I_2}^{I_1} = \exp_{SO(3)}(\boldsymbol{\omega}\Delta t), \tag{17}$$

where Δt is the time interval between the two image frames. Since the device does not provide hardware synchronization between the camera and IMU, we select the IMU values between the two samples with closest timestamps with respect to the two images. By applying rotation matrix $R_{I_2}^{I_1}$ on undistorted point set $\hat{\boldsymbol{p}}_1, \boldsymbol{p}'_2 = R_{I_2}^{I_1}\hat{\boldsymbol{p}}_1$, the relation between $\boldsymbol{p}_1$ and $\hat{\boldsymbol{p}}_2{}'$ would only be a translation vector up to scale

$$\boldsymbol{p}'_2 = \boldsymbol{p}_1 + \mathbf{t},\ \boldsymbol{p}_2'^\top [\boldsymbol{t}]_\times \boldsymbol{p}_1 = 0. \tag{18}$$

Using eq. (18), we can pick at least two corresponding points to estimate the translation vector $\boldsymbol{t} = [t_x, t_y, t_z]$. The estimation procedure is continuously repeated for incoming images keeping I_1 as the keyframe until a new one is selected when the number of tracked features goes below a given value or the yaw angle between two frames is larger than a certain threshold. The outlier rejection step is always performed between two consecutive frames.

D. Drone Detection

In order to select the agent in space with gaze and localize the agent, we trained a deep neural network to detect drones. Our goal is to obtain a real-time solution. For this reason, we decided to use the lightweight yet accurate detection algorithm Single Shot multibox Detector (SSD) described in [37]. The main structure of our network is similar to the original SSD, but with the advantage of having two classes (drone or background) to train. The i^{th} layer of the network produces a feature map of size $h_i \times w_i \times c_i$. For each feature map, the network predicts the objects based on a set of initial bounding boxes called anchor boxes. To customize the network, we set the default number of anchor boxes to be $k = 4$, with aspect ratios (width/height) of $\{1, 2, 3, 4\}$. These default boxes just serve as a set of initial guesses and the network would predict final bounding boxes based on these initial anchor boxes around each location of the feature map. The maximum number of bounding boxes we can have is then $\sum_i k \times h_i \times w_i$. As our ultimate goal is to localize the drone in the image and estimate its depth relative to the camera, we add one more depth regression loss $\mathcal{L}_d$ in eq. (20) to force the network to predict the correct bounding box area. The training objective $\mathcal{L}$ consists of localization loss $\mathcal{L}_l$, confidence loss $\mathcal{L}_c$, which are described in [37], [38] and depth regression loss $\mathcal{L}_d$

$$\mathcal{L} = \frac{1}{N}(\mathcal{L}_c(x, c) + \alpha\mathcal{L}_l(x, p, g) + \beta\mathcal{L}_d(x, c, p, g)), \tag{19}$$

$$\mathcal{L}_d = \sum_{i \in k} \mathbf{1}(x = c)||d_i^w\, d_i^h \exp(p_i^w + p_i^h) - g^w\, g^h||_2, \tag{20}$$

where N is the number of matched proposal boxes, x is the predicted category, c is the ground truth category, p is the

TABLE I
AVERAGE ANGULAR ERROR (AAE) IN RADIAN OF ATTITUDE ESTIMATION WITH ALL 5 SEQUENCES (LEFT) AND RELATIVE POSE ERROR (RPE) IN RADIAN OF GAZE DIRECTION WITH ALL 5 USERS (RIGHT)

	Element	Seq.1	Seq.2	Seq.3	Seq.4	Seq.5	Mean
ESKF	yaw	0.0829	0.1475	0.1564	0.2576	0.1358	0.1564
	pitch	0.0207	0.0456	0.0240	0.0377	0.0205	0.0297
	roll	0.0267	0.0444	0.0203	0.0256	0.0159	0.0266
ESKF+Vision	yaw	0.0187	0.0475	0.0383	0.0825	0.0485	0.0471
	pitch	0.0208	0.0508	0.0305	0.0294	0.0194	0.0302
	roll	0.0267	0.0382	0.0191	0.0295	0.0159	0.0258

	Distance	User 1	User 2	User 3	User 4	User 5	Mean
Static	1m	0.1782	0.1735	0.1683	0.1217	0.1604	0.1604
	2m	0.1363	0.1586	0.1740	0.1721	0.1506	0.15832
	3m	0.1021	0.0904	0.1331	0.1624	0.1340	0.1244
Dynamic	1m	0.1296	0.1701	0.1802	0.1234	0.1021	0.1411
	2m	0.1583	0.1757	0.0710	0.1098	0.1211	0.1271
	3m	0.0909	0.1376	0.1264	0.1652	0.0906	0.1221

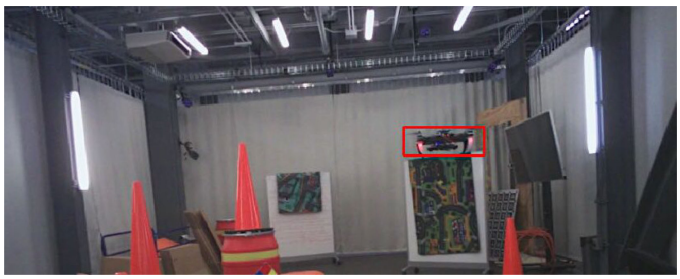

Fig. 3. Our system's detection of a flying quadrotor in the air. Using the parameters output from the network, we draw the red bounding box in the image.

predicted bounding box parameters, g is the ground truth bounding box parameters and d is the default proposal boxes parameters defined in [37]. A sample detection result is shown in Fig. 3. The network is trained on a dataset with 1500 images with size 960×540 pixels each with a ground truth bounding box. We augment the training data by randomly cropping or resizing the images. The Adam optimizer [39] is used with initial learning rate 0.001, which is decayed exponentially by a factor of 0.94 every 2 epochs and clipped at 0.0001 during training. The weight on localization loss, α, is set to be 0.1 and the weight on depth regression loss, β, is set to 0.01. We trained a model with batch size 16 on a 12 GB NVIDIA GeForce 1080 Ti GPU for 150,000 iterations, which takes around 12 hours.

The detection pipeline is not launched until a gaze fixation is detected. We use a dispersion-based algorithm (see, e.g., [40]) that decides a fixation has occurred when the standard deviation of gaze points, σ_g, accumulated during a time window of length Δt, is less than a certain threshold σ_0. Once a fixation is detected, we crop a 300 $\times$ 300 image centered at the gaze position and feed this image into our detector. The cropping step turns out to be important for robust detection as it reduces the image noise and rejects potential false positives. In our experiment, we set σ_0 to be 10 pixels and Δt to be 200 ms. Once the hovering quadrotor is detected, we cross check the bounding box position with the gaze to reject false positive predictions. Given the quadrotor's size H_q, W_q in meters, camera focal length f, and network output bounding box with size h, w in pixels, we compute the human/robot relative depth as

$$d = f\sqrt{\frac{H_q W_q}{hw}} \tag{21}$$

The depth could be underestimated due to a side view of the quadrotor, but this can be easily compensated using the known quadrotor yaw angle from odometry and camera orientation. An alternative method is the additional 3D gaze provided by the manufacturer, which is obtained by triangulating detected pupils on human eyes. However, as pointed out in [41], the accuracy is poor and would require the use of additional external sensor suites for a reliable estimate.

V. EXPERIMENTAL RESULTS

In this section, we report on the experiments performed at PERCH (Penn Engineering Research Collaboration Hub) at the University of Pennsylvania. A Vicon[2] motion capture system is used to report ground truth data at 100 Hz as the attitude estimator and the drone detection algorithm is running at 3 Hz on a NVIDIA Jetson TX2 module. The overall pipeline runs on a portable platform so the user is able to carry the device without requiring a ground station. The top speed of the autonomous agent is set to be 2 m/s.

A. Sensor Characterization

We first perform experiments to characterize the sensor data and demonstrate the accuracy of the proposed technique. In order to study the accuracy of attitude estimation described in Sec. IV-B and Sec. IV-C, we evaluate our algorithm on 5 different sequences. The sequences were recorded with Tobii Glasses worn by 5 different users. Each user was asked to rotate his/her head to search for a virtual target. The sequence duration goes from 23 s to 41 s. We compare our attitude estimation with ground truth obtained from Vicon and report the Average Angular Error (AAE) in radians in Table I (left). From Fig. 4, we can clearly see the yaw estimation with only ESKF drifts over time, while using our method with vision, the yaw drift is reduced and head orientation can be better tracked. Moreover, we would like to study the gaze accuracy with attitude estimation in Table I (right). In the static experiment, a quadrotor hovers in front of the user 1, 2, and 3 meters away. The task for a user is to find and select the hovering agent in the scene using their gaze. We make the user's starting direction randomly deviate ± 90 degrees from the virtual line connecting user and robot. In the dynamic experiment, we have a quadrotor flying at 1 m/s, and users were asked to track the aerial robot with their gaze. Setting the user's starting pose as a reference frame, we calculate the Relative Pose Error (RPE) between the gaze vector $\mathbf{v}_e$ and the ground truth vector $\mathbf{v}_{gt}$ pointing from the user's coordinate origin to the agent's, both in the reference frame. The gaze transformation from local camera frame to reference frame is

[2]www.vicon.com

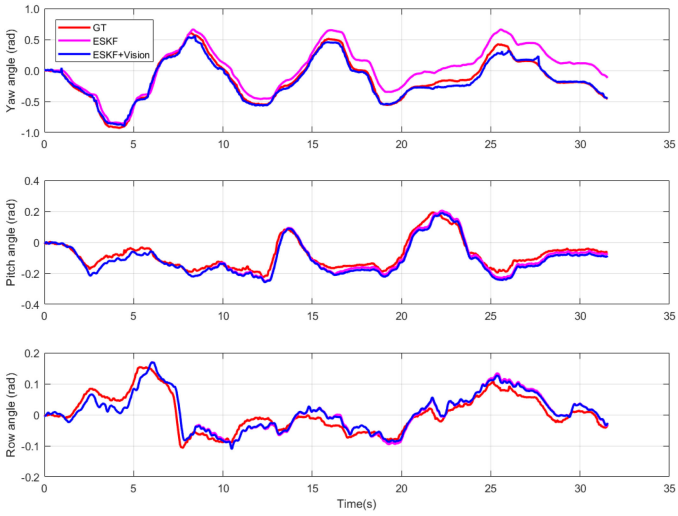

Fig. 4. Attitude estimation in Seq. 3. Ground truth (red), ESKF (magenta), ESKF and vision (blue).

calculated as

$$\lambda \mathbf{v}_e \sim R_t R_c^b K^{-1} \begin{bmatrix} u_g \\ v_g \\ 1 \end{bmatrix}, \tag{22}$$

where R_t is the estimated orientation with respect to the reference frame at time t, R_c^b is the constant rotation matrix that rotates a vector in camera frame to robot body frame, K is the intrinsic matrix of camera and $[u_g\ v_g\ 1]^\top$ is the homogeneous representation of gaze on image. To get the ground truth vector $\mathbf{v}_{gt}$, we use the following equation

$$\mathbf{v}_{gt} = \mathbf{t}_r - \mathbf{t}_g, \tag{23}$$

where $\mathbf{t}_g$ is the glasses position and $\mathbf{t}_r$ is the robot position, all in Vicon coordinates. Finally, the RPE is calculated as

$$RPE = \arccos\left(\frac{\mathbf{v}_e \cdot \mathbf{v}_{gt}}{||\mathbf{v}_e||_2 ||\mathbf{v}_{gt}||_2}\right), \tag{24}$$

Again, in the static test, we decide that the user's gaze is "fixed" when the gaze's standard deviation is less than 10 pixels within 200 ms time window and report error metrics using gaze data outside this window. In the dynamic test, we use the overall data sequence to compute the error metrics. In both static and dynamic experiments, we ask users to conduct tasks within a period of time and experiments took 10 to 15 s. From experiment results, we observe that the mean RPE is, on average, lower for dynamic case comparing to the static ones. We believe this is due to the fact that it is more natural for a human to move his or her eyeballs to track a dynamic object than to fix his/her gaze in space. A 15 s tracking task seems to be easy for human beings, but fixating gaze at one point was more challenging. In Fig. 5, we show the error curve in one of the dynamic sequences. We hypothesize that the periodic fluctuations coincide with the user blinking or loosing attention.

B. Drone Detection and Localization

Our drone detector serves two purposes. First, the detections are used along with the gaze to enable agent selection. Second, the detections are used to estimate agent's depth. In this section, we report the Root Mean Squared Error (RMSE) of the depth estimation compared to the ground truth depth obtained from the Vicon motion capture in each sequence. The estimated depth was calculated from eq. (21) and the ground truth depth was obtained by using eq. (23) and converting the resulting 3D vector into the glasses body frame. In every sequence, the quadrotor hovers in front of the user, and we ask the user to try to gaze only at the robot. Multiple detections are triggered according to our policy described in Section IV-D. We study the different distances and report the error metrics in Table II considering over 90 frames per sequence. The results show competitive depth estimation from our learning-based method. We can clearly observe that using gaze to propose a region of interest (ROI) for the network provides robust detection, especially for objects that are distant from the user. As a baseline, we have also evaluated a color-based method to localize the robot on the same dataset. By applying ellipse fitting on color segmentation contours and estimating the ellipse major axis, we can estimate the relative depth, similar to our previous work [42]. We noticed that the quality of detection provided by this method depends excessively on the color segmentation, which is sensitive to light conditions. This strongly affects the depth estimation, resulting in larger errors than those induced by our proposed method. We believe that more advanced learning-based algorithms, such as Faster R-CNN or Mask R-CNN, would improve localization performance. However, better performance would be obtained at the price of an increased computation time preventing real-time operation.

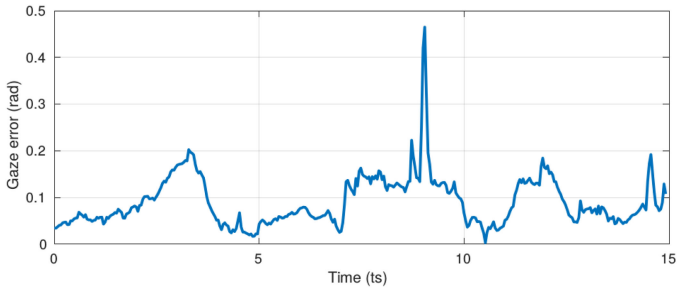

Fig. 5. Gaze error curve in 3 meters dynamic test of user 5.

TABLE II
ROOT MEAN SQUARE ERROR (RMSE) AND STANDARD DEVIATION (STD) OF DEPTH ESTIMATION IN METERS WITH THE CORRESPONDING NUMBER OF FRAMES USED FOR EVALUATION

	distance	1 m	2 m	3 m
	# of frames	106	122	91
Ours	RMSE	0.109	0.125	0.297
	STD	0.054	0.074	0.106
Color-based	RMSE	0.524	0.594	0.953
	STD	0.447	0.440	0.810

C. Gaze-Assisted Autonomous Navigation

We test the overall system, i.e., human-assisted quadrotor navigation. After the quadrotor is identified, the human can then fixate on different locations in space to spatially task the

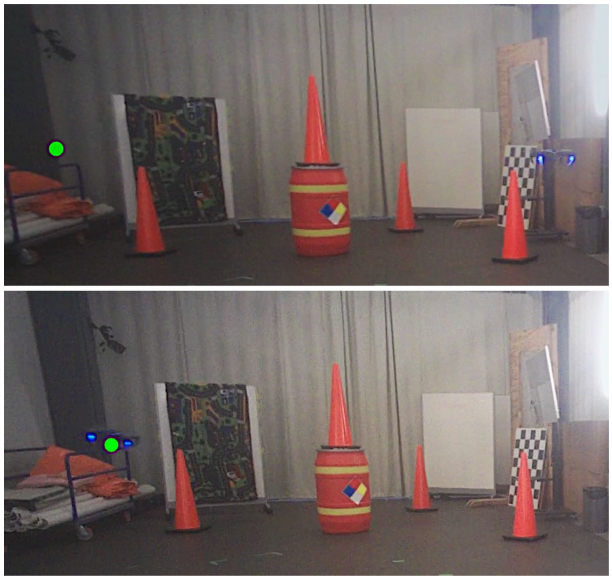

Fig. 6. Gaze-based quadrotor spatial tasking (user view). The user fixates a spatial location (top), denoted with the green dot, and the drone plans a straight path to reach it (bottom).

platform. Technically, if communication is available between the human and the robot, the human can detect the robot just when performing the first interaction so further realignment is not strictly necessary. It can be re-detected to compensate for drifts related to the aerial robot odometry or if the communication from the robot to the human drops. Once the relative position is known, the human can send commands directly in the local frame of the robot. To compute the 3D navigation waypoint, we use the 2D gaze coordinate provided from the glasses to compute a pointing vector from the glasses, and select the waypoint depth within a predefined safety zone. Ideally, the 3D navigation waypoint would come directly from the eye tracking glasses, but we found in our experiments that the depth component reported by the glasses was too noisy to be used. In the future, we plan to further investigate this issue in order to give the user more depth control. Nevertheless, in the attached multimedia material, we show that it is possible for a human-user to move the quadrotor in space by fixating his/her gaze on a specific location. This is depicted in Fig. 6, where the robot executes a maneuver to position itself along the human-gaze direction. Multiple tests are performed in which the human attempts to task the robot to go to different areas using combinations of both head and gaze movement. The robot localization and tracking performances have been evaluated in our previous contribution [30] for multiple trajectories and are equivalent in this case.

D. Limitations and Extensions

The proposed approach is a first step toward a new form of non-invasive and intuitive interaction between humans and robots. The current sensor-processing solution, despite being tested on a few users, aims to present the methodology to enable gaze-based control, though a more extensive user-study is planned for the future work. Moreover, while the system should not be considered as a monolithic solution to the problem of human-robot interaction, it can complement and augment other interaction modalities to create complex and more meaningful ways of data interpretation. In fact, multiple studies [12], [13] suggest that gaze can be used as a flexible cue for eliminating uncertainty and ambiguity about referential expressions. The introduction of new modalities of interaction can, for example, address the limitations of previous works, which may, for example, require the vehicle to be in the user's field of view when moving in space. The different modalities can be incorporated in the proposed Kalman filter to predict different objects the human wants to interact with or to select the vehicle, speeding up the human-robot collaborative task and refining user's intention interpretation.

In addition, the proposed solution can also help the creation of a drone companion [43], helping people affected by mobility issues such as those with paraplegia. In these situations, the person can take advantages of the playful nature of the system and direct the robot with gaze, and perhaps a verbal cue, to make an observation of a target or to obtain advantageous observation points of the environment. The system will also enable people with very little drone experience to safely and effectively fly drones in situations where finding a dedicated pilot is not be possible. Finally, the navigation strategy can be extended including MAV yaw control. It is natural to assign a spatial location to the robot's position, but it is more complex to command the yaw. A simple and intuitive way would be to select it to have the robots heading always pointing according to the trajectory tangent direction. This can relax the current implicit assumption of navigation in free space facilitating the identification of obstacles located in front of the robot. In this way, if the initial straight path is occluded, the vehicle can reach the final destination by re-planning. Another possibility is to use the yaw to keep the user in the field of view of the robot camera if the user focuses his visual attention on a concurrent task, or to map other forms of interactions to yaw motions.

VI. Conclusion

In this work, we presented a first step toward enabling human-MAV teams for spatial tasking by utilizing human eye gaze. The proposed approach is able to decouple head orientation from eye gaze, while concurrently identifying the aerial vehicle. Our solution is portable and allows spatial control of the aerial platform by gazing at specific locations without requiring the user to be in the robot's field of view.

We discussed the advantages, limitations, and extensions of the approach, which involve increased control and fusion with multi-modal interaction types. We would also like to explore the ability to consider multiple agents to obtain a complete multi-human/multi-robot collaborative system. Finally, we would like to investigate the need of providing users with a feedback about detected waypoints despite the current solution directly allows to verify if the robot has reached the goal. We believe our solution opens up new ways to interpret human attention and create new anticipative human-robot interfaces.

REFERENCES

[1] T. Tomic *et al.*, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[2] T. Ozaslan *et al.*, "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1740–1747, Jul. 2017.

[3] G. Loianno, J. Thomas, and V. Kumar, "Cooperative localization and mapping of MAVs using RGB-D sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 4021–4028.

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

[5] N. Michael *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *J. Field Robot.*, vol. 29, no. 5, pp. 832–841, 2012.

[6] "Tobii pro glasses 2 product description," [Online]. Available: https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/. Accessed: Sep. 7, 2018.

[7] A. J. Calder *et al.*, "Reading the mind from eye gaze," *Neuropsychologia*, vol. 40, no. 8, pp. 1129–1138, 2002.

[8] C. Teufel, P. C. Fletcher, and G. Davis, "Seeing other minds: Attributed mental states influence perception," *Trends Cogn. Sci.*, vol. 14, no. 8, pp. 376–382, 2010.

[9] C. Teufel, D. M. Alexis, N. S. Clayton, and G. Davis, "Mental-state attribution drives rapid, reflexive gaze following," *Attention, Perception, Psychophysics*, vol. 72, no. 3, pp. 695–705, 2010.

[10] C. Yu, P. Schermerhorn, and M. Scheutz, "Adaptive eye gaze patterns in interactions with human and artificial agents," *ACM Trans. Interact. Intell. Syst.*, vol. 1, no. 2, pp. 13:1–13:25, Jan. 2012.

[11] M. F. Land and M. Hayhoe, "In what ways do eye movements contribute to everyday activities?" *Vision Res.*, vol. 41, no. 25, pp. 3559–3565, 2001.

[12] J. E. Hanna and S. E. Brennan, "Speakers eye gaze disambiguates referring expressions early during face-to-face conversation," *J. Memory Lang.*, vol. 57, no. 4, pp. 596–615, 2007.

[13] M. Staudte and M. W. Crocker, "Investigating joint attention mechanisms through spoken human–robot interaction," *Cognition*, vol. 120, no. 2, pp. 268–291, 2011.

[14] H. Admoni and B. Scassellati, "Social eye gaze in human-robot interaction: A review," *J. Hum.-Robot Interact.*, vol. 6, no. 1, pp. 25–63, May 2017.

[15] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. Di Caro, "Human-swarm interaction using spatial gestures," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3834–3841.

[16] K. Pfeil, S. L. Koh, and J. LaViola, "Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles," in *Proc. Int. Conf. Intell. User Interfaces*, 2013, pp. 257–266.

[17] W. S. Ng and E. Sharlin, "Collocated interaction with flying robots," in *Proc. RO-MAN*, 2011, pp. 143–149.

[18] M. Walker, H. Hedayati, J. Lee, and D. Szafir, "Communicating robot motion intent with augmented reality," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact.*, 2018, pp. 316–324.

[19] O. Erat, W. A. Isop, D. Kalkofen, and D. Schmalstieg, "Drone-augmented human vision: Exocentric control for drones exploring hidden areas," *IEEE Trans. Visualization Comput. Graph.*, vol. 24, no. 4, pp. 1437–1446, Apr. 2018.

[20] C. Reardon, K. Lee, and J. Fink, "Come see this! augmented reality to enable human–robot cooperative search," in *Proc. IEEE Symp. Safety, Secur. Rescue Robot.*, Aug. 2018, pp. 1–7.

[21] J. Kennedy, P. Baxter, and T. Belpaeme, "Head pose estimation is an inadequate replacement for eye gaze in child-robot interaction," in *Proc. 10th ACM/IEEE Int. Conf. Human-Robot Interact. Extended Abstr.*, 2015, pp. 35–36.

[22] E. G. Freedman and D. L. Sparks, "Coordination of the eyes and head: Movement kinematics," *Exp. Brain Res.*, vol. 131, no. 1, pp. 22–32, Mar. 2000.

[23] M. Yu, Y. Lin, D. Schmidt, X. Wang, and Y. Wang, "Human–robot interaction based on gaze gestures for the drone teleoperation," *J. Eye Movement Res.*, vol. 7, no. 4, pp. 1–14, 2014.

[24] J. P. Hansen, A. Alapetite, I. S. MacKenzie, and E. Møllenbach, "The use of gaze to control drones," in *Proc. Symp. Eye Tracking Res. Appl.*, 2014, pp. 27–34.

[25] Y.-S. Jiang, G. Warnell, E. Munera, and P. Stone, "A study of human–robot copilot systems for en-route destination changing," in *Proc. IEEE Int. Symp. Robot Human Interactive Commun.*, Aug. 2018, pp. 997–1004.

[26] Y.-S. Jiang, G. Warnell, and P. Stone, "Inferring user intention using gaze in vehicles," in *Proc. 20th ACM Int. Conf. Multimodal Interact*, 2018, pp. 298–306.

[27] J. Bruce, J. Perron, and R. Vaughan, "Ready—aim—fly! hands-free face-based HRI for 3D trajectory control of UAVs," in *Proc. 14th Conf. Comput. Robot Vision*, 2017, pp. 307–313.

[28] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella, "Human control of UAVs using face pose estimates and hand gestures," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact.*, 2014, pp. 252–253.

[29] V. M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori, "HRI in the sky: Creating and commanding teams of UAVs with a vision-mediated gestural interface," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 617–623.

[30] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 404–411, Apr. 2017.

[31] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2016, pp. 4304–4311.

[32] A. Santamaria-Navarro, G. Loianno, J. Solà, V. Kumar, and J. Andrade-Cetto, "Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors," *Auton. Robots*, vol. 42, no. 6, pp. 1263–1280, Aug. 2018.

[33] F. Fraundorfer, P. Tanskanen, and M. Pollefeys, "A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 269–282.

[34] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vision*, May 2006, vol. 1, pp. 430–443.

[35] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 1994, pp. 593–600.

[36] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, 1994.

[37] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[39] J. B. Diederik and P. Kingma, "Adam: A method for stochastic optimization," Dec. 2014, arXiv preprint arXiv:1412.6980.

[40] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proc. Symp. Eye Tracking Res. Appl.*, 2000, pp. 71–78.

[41] H. Wang, J. Pi, T. Qin, S. Shen, and B. E. Shi, "Slam-based localization of 3D gaze using a mobile eye tracker," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, 2018, pp. 65:1–65:5.

[42] R. Tron, J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "A distributed optimization framework for localization and formation control: Applications to vision-based measurements," *IEEE Control Syst. Mag.*, vol. 36, no. 4, pp. 22–44, Aug. 2016.

[43] K. D. Karjalainen, A. E. S. Romell, P. Ratsamee, A. E. Yantac, M. Fjeld, and M. Obaid, "Social drone companion for the home environment: A user-centric exploration," in *Proc. 5th Int. Conf. Human Agent Interact.*, 2017, pp. 89–96.