Collaborative Control of Aerial Robots for Inferring Human Intent from Gaze Following

Vishnu S. Chipade[1], Alia Gilbert[2], Daniel Harari[3], Dimitra Panagou[2,1]

***Abstract*— In an ideal human-robot collaboration, autonomous robots work side-by-side with humans in a joint workspace, often performing complementary tasks to the humans. A robotic ability to infer human intention and goals directly from human behavior will facilitate the collaboration and maximize its efficiency. In this paper, we focus on inferring which object the human wants picked up next, based on what the human is looking at, by visually following the human gaze and head orientation. We develop a coordination protocol for a team of aerial robots to extract effective human head and gaze cues. The aerial robots are controlled to navigate around the human and collect data that improves the detection of the human's gaze and hence the intended object to be picked up. The effectiveness of the approach is shown using simulations in AirSim, a photo-realistic simulator.**

I. Introduction

With autonomous systems heavily integrated into everyday life, human-robot collaboration has become an important area of research in the last decade [1]. Effective collaboration dictates effective communication between humans and robots. Humans have developed highly-effective communication abilities among themselves, including non-verbal means, mainly relying on visual cues. Research in recent years has focused on how agents (e.g., robots) can analyze visual cues including body pose [2], face [3] and gaze direction [4] [5] to interpret human intent. In certain cases, humans can communicate their intentions to others using other means of communication including verbal description or hand gestures. However, these means of communication require more effort and are often provided only on-demand. Since we would like the robot to operate in a seamless way and fit-in with the human common and natural behavior, gaze following is the best cue for inferring human intent; gaze cues are usually available earlier, and may be more reliable than other cues since the human's own planning, including navigation, motion and interactions, is based on gaze locking at the target.

Methods for gaze and head-pose estimation [6] aim to improve human-robot collaboration [7]–[9]. Often the human is asked to wear gaze-tracking devices [10], [11]. While typically of high precision, these devices are not always practical as the equipment can be prohibitively expensive, or restrict the human's motion and behavior.

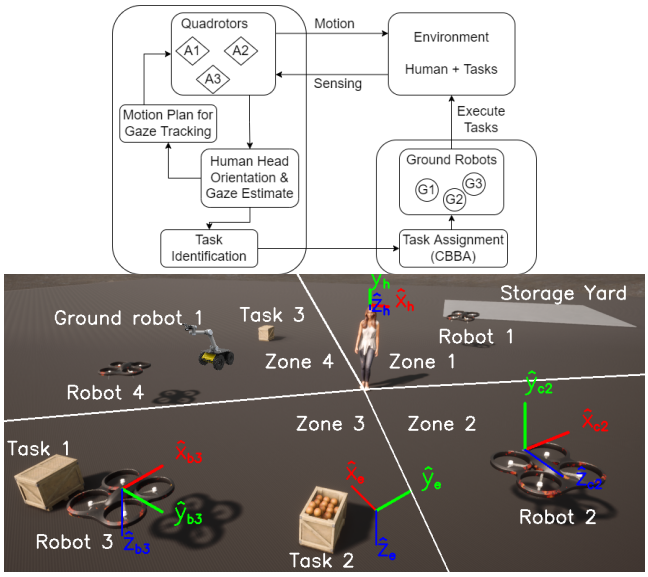


Fig. 1: Problem setup and proposed architecture (The zones here are just representative, for details on how these zones are defined refer to Section IV and Fig. 3a)

In contrast to the above approaches, in this paper, we seek to utilize aerial robots with onboard cameras around the human to extract estimates of the human gaze. To this end, we consider that current visual-gaze estimation models require that the face is visible in the camera's field-of-view. We envision the development of a heterogeneous multi-robot system that aims to seamlessly operate near humans at work, observe them and infer their intent, and automatically provide assistance when needed. In this paper, we focus on developing a collaborative controller for aerial robots that operate around the human so that at least one of them tracks the human face at all times, extracts body and facial features, and uses these features to follow gaze direction and infer human intent. This information is then used by ground robots to assist the human with their tasks. *The contribution of this paper is particularly the collaborative control algorithm under which the aerial robots navigate around the human so that they improve the accuracy of gaze estimation as obtained by XGaze [12], a state-of-the-art gaze estimation algorithm.*

More specifically, in our current simulation setting we consider a system of heterogeneous robots, namely:

[1] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA; [2] Department of Robotics, University of Michigan, Ann Arbor, MI, USA; [3] Weizmann AI Center, Department of Computer Science and Applied Math, Weizmann Institute of Science, Israel; vishnuc@umich.edu, galia@umich.edu, hararid@weizmann.ac.il, dpanagou@umich.edu

1) aerial robots equipped with RGB-D sensors, and 2) ground robots equipped with capability to manipulate objects of certain weights and sizes, which assist humans with lifting and moving them to a storage location. Such system could be useful in farms and storage houses, where humans may need assistance with lifting heavy objects, and where deployment of fixed cameras might be impossible or inefficient (e.g., in large agricultural fields). In the specific setting considered in this paper, a human is moving in an environment populated with objects and looks at the intended (target) object while approaching it. The aerial robots use the information from the visual sensors to follow the human and always keep the face visible by at least one robot. Having the human face always in sight allows the system to infer the location of the target object by following the human's gaze direction.

To achieve this, the aerial robots are assigned operational zones (gray zones in Fig. 3a) that are fixed with respect to the human's position. A favorable zone (blue zone in Fig. 3a) is created around the human, based on the reliability of gaze estimation (using XGaze [12], a pretrained algorithm) as a function of the human-head orientation. This favorable zone rotates as the human turns while moving in the environment. Each aerial robot moves around the human while remaining in its operational zone, and while switching between the following two modes: the *face-tracking mode*, when the favorable zone of the human intersects with the operational zone of the robot so that the front of the human face is visible to the robot, and the *body-tracking mode*, otherwise. In face-tracking mode, the aerial team moves such that the human always faces one of the robots, allowing an improved gaze estimation capability from the frontal view. The aerial robot that is operating in *face-tracking mode* may keep changing as the human moves and turns around. In the body-tracking mode, each robot tries to keep the entire human body in the field-of-view. To enable the face-tracking and body-tracking modes, we design the control scheme using potential function based approach ([13], [14]) for the aerial robots so that at least one is in face-tracking mode at all times, collect video and infer the human's gaze direction, and based on that infer which tasks (target objects) in the environment are under observation by the human. A list of tasks is created based on the inferred human intent, and this list is communicated to the ground robots for further assistance. The grounds robots utilize a consensus-based bundle algorithm (CBBA) [15] to asynchronously assign tasks among themselves as new tasks are being generated by the human moving around. In our experiments, we use AirSim, a photorealistic, physics-based simulator for drones [16] and an environment with animated human character in Unreal Engine. The overall human-robot collaboration framework is shown in Figure 1.

The contributions of this paper include: 1) A collaborative control scheme for a team of aerial robots to safely navigate around a human, while keeping the face in sight for accurately extraction of the human gaze detection, 2) a demonstration of a human-robot collaboration consisting of a team of heterogeneous robots in the photorealistic simulation environment AirSim [16].

The rest of the paper is organized as follows: Section II describes the mathematical modeling and problem statement. The main trajectory optimization formulation is discussed in Section V, while simulation based results are provided in Section VI. The conclusions and the ongoing work are discussed in Section VII.

II. Modeling and Problem Statement

Notation:$\|\cdot\|$ denotes Euclidean norm of its argument. $|\cdot|$ denotes absolute value of a scalar argument, and cardinality if the argument is a set. We denote by $\hat{\mathbf{x}} = [1,0,0]^T$, $\hat{\mathbf{y}} = [0,1,0]^T$ and $\hat{\mathbf{z}} = [0,0,1]^T$ the Euclidean unit vectors.

We consider a human moving around some object locations (called thereafter tasks) on the ground, N_a aerial robots denoted by $\mathcal{A}_i$ for $i \in I_a = \{1,2,...,N_a\}$ in the human's neighborhood, and N_g ground robots denoted by $\mathcal{G}_j$ for $j \in I_g = \{1,2,...,N_g\}$. An RGB-D sensor is rigidly fixed on each robot to point towards the forward-looking direction. We consider the following right-handed coordinate frames (Figure 1): 1) An earth-fixed coordinate frame, $\mathcal{F}_e$, formed by $\hat{\mathbf{x}}_e$, $\hat{\mathbf{y}}_e$, $\hat{\mathbf{z}}_e$ axes pointing in local North, East and Down directions; 2) A human-head fixed coordinate frame, $\mathcal{F}_h$, formed by $\hat{\mathbf{x}}_h$, $\hat{\mathbf{y}}_h$, $\hat{\mathbf{z}}_h$ axes, where $\hat{\mathbf{z}}_h$ points in the front direction of the human face and $\hat{\mathbf{x}}_h$ points towards the left direction of the human face; 3) a body fixed coordinate frame for robot $\mathcal{A}_i$, $\mathcal{F}_{b_i}$, formed by $\hat{\mathbf{x}}_{b_i}$, $\hat{\mathbf{y}}_{b_i}$, $\hat{\mathbf{z}}_{b_i}$ axes, where $\hat{\mathbf{x}}_{b_i}$ and $\hat{\mathbf{y}}_{b_i}$ point in the front and right direction of the robot $\mathcal{A}_i$, respectively; 4) a camera fixed coordinate frame for robot $\mathcal{A}_i$, $\mathcal{F}_{c_i}$, formed by $\hat{\mathbf{x}}_{c_i}$, $\hat{\mathbf{y}}_{c_i}$, $\hat{\mathbf{z}}_{c_i}$ axes, where $\hat{\mathbf{x}}_{c_i}$ and $\hat{\mathbf{z}}_{c_i}$ point in the right direction and back of the robot $\mathcal{A}_i$, respectively (see Figure 1).

The pose of the human head, as resolved in $\mathcal{F}_e$, is denoted by $\mathbf{r}_h = [\mathbf{p}_h^T, \mathbf{o}_h^T]^T$, where $\mathbf{p}_h = [x_h, y_h, z_h]^T$ and $\mathbf{o}_h = [\phi_h, \theta_h, \psi_h]^t$ are the position and the orientation of the human head, respectively, where ϕ_h, θ_h, ψ_h are the rotations about the axes $\hat{x}_h$, $\hat{y}_h$, $\hat{z}_h$, respectively, as per $z(\psi_h) - y(\theta_h) - x(\phi_h)$, i.e., 3-2-1 intrinsic Euler rotation. The linear speed of the human head in $\mathcal{F}_e$ is denoted as $\mathbf{v}_h = [v_{x,h}, v_{y,h}, v_{z,h}]^T$ and the angular velocity as resolved in $\mathcal{F}_h$ is denoted as $\boldsymbol{\omega}_h = [\omega_{x,h}, \omega_{y,h}, \omega_{z,h}]^T$. We denote by $\mathbf{r}_i = [\mathbf{p}_i^T, \psi_i]^T$ the controllable states of the robot $\mathcal{A}_i$ and consequently of the mounted RGB-D sensor, where $\mathbf{p}_i = [x_i, y_i, z_i]^T$ denotes the position of the robot in $\mathcal{F}_e$, and ψ_i is the yaw angle of $\mathcal{A}_i$.

The motion of the robots is modelled as:

$$\dot{\mathbf{r}}_i = \begin{bmatrix} \dot{\mathbf{p}}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} \mathbf{v}_i \\ \omega_{z,i} \end{bmatrix} \tag{1}$$

for all $i \in I$, where $\mathbf{v}_i$ and $\omega_{z,i}$ are the translational velocity and the yaw rate of the robot $\mathcal{A}_i$, which also act as control inputs. We also assume that $\|\mathbf{v}_i\| < \bar{v}$, for all

$i \in I$, where $\bar{v} > 0$ is the maximum speed of the robots. In addition, we make the following assumptions.

***Assumption** 1:* The roll ϕ_i and pitch θ_i angles of the robots during the translational motion are very small.

***Assumption** 2:* The linear speed $\mathbf{v}_h$ of the human is bounded as $\|\mathbf{v}_h\| \leq \bar{v}_h$, and the rotational speed of the human head is bounded by $\|\boldsymbol{\omega}_h\| \leq \bar{\omega}_h$.

***Assumption** 3:* The current position $\mathbf{p}_h$ and the linear velocity $\mathbf{v}_h$ of the human head at any instant is known to all the robots.

***Assumption** 4:* The robots can move faster than the human, i.e., $\bar{v}_h < \bar{v}$, to ensure that they can keep track of a moving human.

Assumption 1 ensures that each robot $\mathcal{A}_i$ does not tilt too much during its motion and hence the images obtained by the on-board sensor are very close to that obtained by an RGB sensor at $\mathbf{r}_i$ with yaw angle ψ_i with zero pitch and zero roll angle. The aerial robots are deployed in Unreal Environment using AirSim plugin. The human is also deployed in Unreal and is controlled using Unreal Engine's animation tools. See Figure 1 for a snapshot of the simulation scene in Unreal.

***Problem** 1 (Motion Control for Gaze Detection):* Design a control algorithm for a team of N_a aerial robots mounted with RGB-D sensor such that: 1) both eyes, the face and the front body of the human is visible to at least one RGB-D sensor so that human gaze can be detected as accurately as possible, 2) the robots stay safe from each other and the human while moving around, i.e., $\|\mathbf{p}_i(t) - \mathbf{p}_{i'}(t)\| > \rho_{safe}$, for all $i \neq i' \in I_a$ and $\|\mathbf{p}_i(t) - \mathbf{p}_h(t)\| > \rho_{safe}$ for all t.

III. Estimation of Head Orientation and Gaze

In this section, we describe the algorithms that we used to estimate head orientation and gaze direction from video images captured by the onboard cameras of the robots. Then, we describe how the estimated head pose and gaze directions are related to the system's coordinate frames for motion planning (as described in section V).

1) Human head orientation: We used a deep neural network model based on a common architecture [17], [18] to estimate the head orientation in RGB images [19]–[21]. In particular, we used *deep-head-pose* network model [22], to predict intrinsic 3D Euler angles corresponding to the head rotation with respect to the camera frame, directly from image intensities. The model is applied to image regions containing human faces, which are extracted by a face detection neural network model [23]. In our experiments, we used network models that were pre-trained on a large, synthetically-expanded image dataset of faces in a wide range of orientations [24].

2) Human gaze: Gaze estimation is more challenging than estimating the head orientation: it requires the acquisition of fine and subtle details in the face appearance (such as the eye's position in the orbits), and associating them to a direction in space towards the gaze target [25]. In this work we used the *XGaze* deep neural network model [12] that was pre-trained on a large dataset of a wide range of unconstrained face images [26].

3) Associating head and gaze directions to the robots: The head orientation and gaze direction algorithms are applied to images acquired by the camera of each aerial robot. The algorithm outputs are rotation matrices $R_{c_i h}$ and $R_{c_i g}$, corresponding to the head orientation and gaze direction respectively, in the head frame $\mathcal{F}_h$ relative to the camera frame $\mathcal{F}_{c_i}$ of the robot $\mathcal{A}_i$. In the remainder of this section, we will focus on the head rotation matrix $R_{c_i h}$, but the same also applies to the gaze rotation matrix $R_{c_i g}$.

The rotation matrix corresponding to the rotation of the human head relative to the earth fixed frame, R^i_{eh}, based on the measurement from robot $\mathcal{A}_i$, is:

$$R^i_{eh} = R_{eb_i} R_{b_i c_i} R_{c_i h}. \tag{2}$$

where R_{eb_i} is the rotation matrix corresponding to the 3-2-1 intrinsic Euler rotation from Earth frame to the body frame of aerial robot $\mathcal{A}_i$, and $R_{b_i c_i} = \begin{bmatrix} \hat{y} & -\hat{z} & -\hat{x} \end{bmatrix}$.

The unit vector $\hat{\mathbf{z}}^i_h$ pointing in the front of the human head, called human head vector, based on the estimated rotation R^i_{eh} is then obtained as:

$$\hat{\mathbf{z}}^i_h = R^i_{eh} \hat{\mathbf{z}}. \tag{3}$$

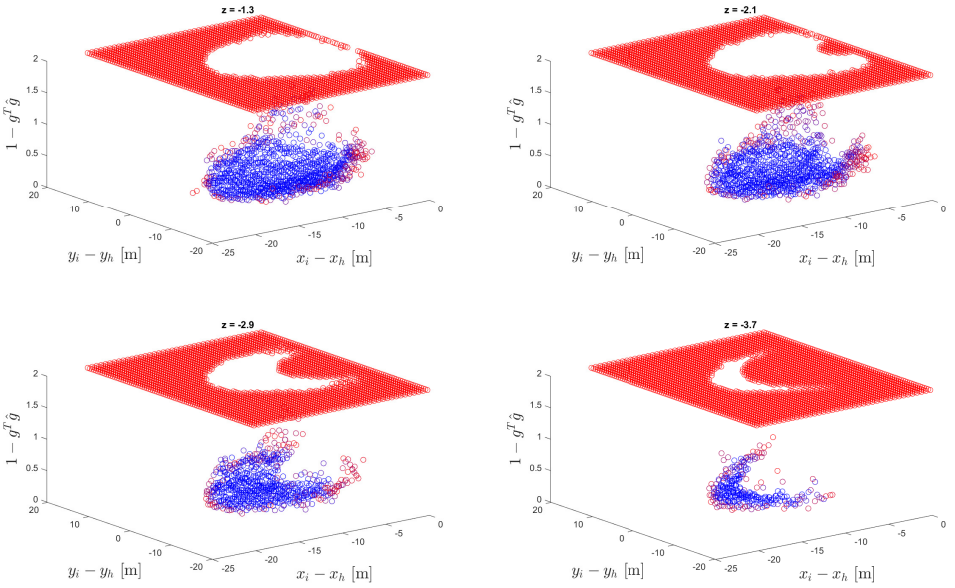
The quality of this estimate depends on the distance of the camera to the human, and the direction from which the camera is looking at the human head. Since we have more than one estimates of $\hat{\mathbf{z}}^i_h$ and probably with varying accuracy from multiple robots, we use a weighted sum as our final estimate of human head vector as:

$$\hat{\mathbf{z}}_h = \frac{1}{|\mathcal{I}_h|} \textstyle\sum_{i \in \mathcal{I}_h} w_i \hat{\mathbf{z}}^i_h \tag{4}$$

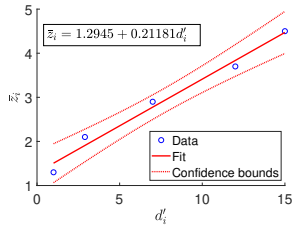
where $\mathcal{I}_h$ is the set of indices of the robots that have the human's face detected in their captured image, and $w_i = \frac{O_i(\mathbf{r}_h)}{\sum_{i \in \mathcal{I}_h} O_i(\mathbf{r}_h)}$ is the weight of the robot $\mathcal{A}_i$'s estimate, where $O_i(\mathbf{r}_h) = \frac{(\hat{\mathbf{z}}^-_h)^T (\mathbf{p}_i - \mathbf{p}_h)}{d_i}$ is the cosine of the angle made by the position vector of the robot $\mathcal{A}_i$ relative to human head and the estimate $\hat{\mathbf{z}}^-_h$ of the human head vector prior to the current measurements, and $d_i = \|\mathbf{p}_i - \mathbf{p}_h\|$ is the distance between $\mathcal{A}_i$ and the human head. The weight w_i is chosen as above to trust more the estimate of the robots whose camera axis is closely aligned with the human head vector. This is because the estimates from such robots are likely more accurate because the front of the face is clearly visible in the captured image. We denote by $\mathbf{o}_{h,m}$ the measurement of the Euler angles of the head extracted from the rotation matrix R_{eh} as:

$$\mathbf{o}_{h,m} = [\phi_{h,m}, \theta_{h,m}, \psi_{h,m}]^T = \text{rot2euler}(R_{eh}) \tag{5}$$

where rot2euler is the function that outputs the Euler angles for 3-2-1 intrinsic rotation corresponding to the rotation matrix R_{eh}.



(a) Human gaze error vs camera position



(b) Maximum height of the camera vs planar distance for successful face detection

Fig. 2: Human gaze error and characterization of camera height limits

IV. Favorable zone for better Gaze estimation

The performance of the *XGaze* algorithm [12] depends on the perspective from which the face is captured, and on how many pixels cover the eyes in the image. These performance-influencing aspects are tightly connected to the camera parameters, scene backgrounds, and the training data used to train the XGaze algorithm [12].

Given an already trained XGaze algorithm, our first goal is to find favorable zones for the aerial robots from where the gaze detection performance is maximized. To do so, we first created a synthetic data-set using AirSim. This data-set consists of: 1) images captured by the camera onboard the aerial robot by moving the aerial robot at different locations on a user-defined grid in front of the human, 2) ground-truth gaze direction ($\mathbf{g}_h$) of the human from Unreal Environment. Let the gaze error be:

$$e_{gaze} = 1 - \mathbf{g}_h^T \hat{\mathbf{g}}_h, \tag{6}$$

where $\mathbf{g}_h$ is the unit vector pointing into the direction of the human's gaze, called as gaze vector, and $\hat{\mathbf{g}}_h$ is an estimate of the gaze vector obtained from XGaze algorithm. For a human with a constant gaze and with a face pointed in negative x direction of the earth frame, the gaze error e_{gaze} as a function of the relative position of the camera in the x-y plane is shown in Figure 2a for different camera heights. As observed in Figure 2a, the gaze error is very small within a certain distance from the human and when the camera is directly in front of the human. As the height of the camera is increased, the minimum distance from which the face of the human is detected is also increased. This also means that for a given planar distance $d_i' = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2}$ from the human, the camera has to stay below a certain height $\bar{z}_i$ to ensure successful detection of the face and in turn successful gaze estimation. Using least square regression, we fit a linear curve, shown in Figure 2b as the solid red line, whose parameters are the slope m_g and the y-intercept c_g, to the maximum height $\bar{z}_i$ and the planar distance d_i' data points from Figure 2b.

The maximum limit on the height of the sensor (and in turn that of the robot) is then obtained to be:

$$-z_i < \bar{z}_i = c_g + m_g d_i' \tag{7}$$

where the intercept and slope parameters are computed as $c_g = 1.2945$ and slope $m_g = 0.21181$, respectively. This upper limit on the height results into a favorable zone $\mathcal{Z}_f$ for the aerial robots given by

$$\mathcal{Z}_f(\mathbf{r}_h) = \{\mathbf{p}_i \in \mathbb{R}^3 | d_i' < \bar{d}_i',\ \hat{\mathbf{z}}_h^T(\Delta\mathbf{p}') > 0, z_i \in (0, -\bar{z}_i)\} \tag{8}$$

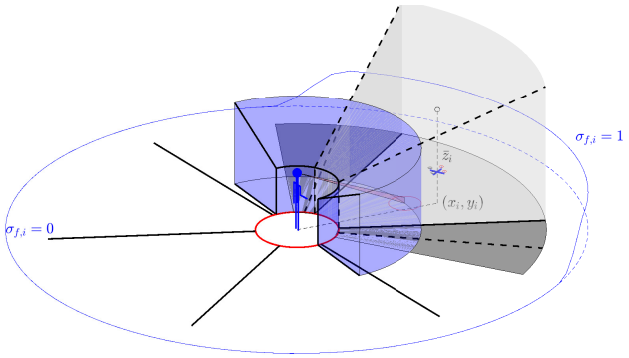
where $\Delta\mathbf{p}' = \mathbf{p}_i' - \mathbf{p}_h'$, $\mathbf{p}_i' = [x_i, y_i, 0]^T$ and $\mathbf{p}_h' = [x_h, y_h, 0]^T$, and the maximum planar distance $\bar{d}_i' = 15$ m. This favorable zone $\mathcal{Z}_f$ rotates as the human rotates, and two instances of it are shown in Figure 3a and 3b. In the next section, we design a motion planning algorithm so that at least one of the aerial robots stays in the favorable zone $\mathcal{Z}_f$ and as much in the front of the human as possible.

V. Motion Planning for the Robots

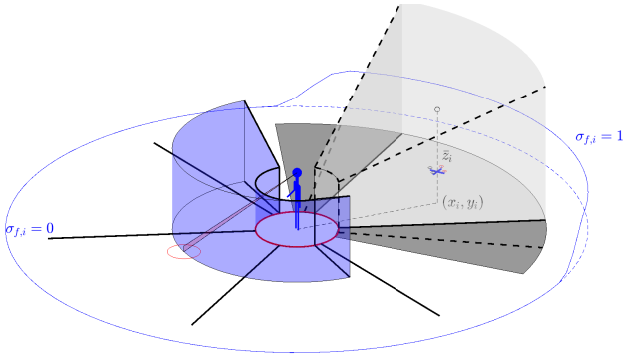
Each robot $\mathcal{A}_i$ operates in their assigned zone $\mathcal{Z}_i$, called as operational zone, see Figure 3a, defined as:

$$\mathcal{Z}_i(\mathbf{r}_h) = \{\mathbf{p}_i \in \mathbb{R}^3 | -\bar{z}_i < z_i < 0, \underline{\psi}_{oz,i} \leq \psi_{h,i} < \bar{\psi}_{oz,i}\} \tag{9}$$

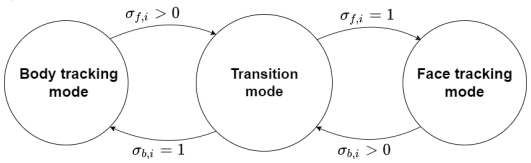
where $\psi_{h,i} = \tan^{-1}(\frac{y_i - y_h}{x_i - x_h})$. In their respective zones, as required in Problem 1, the robots have to ensure



(a) Favorable zone (blue) and $\mathcal{A}_i$'s operational zone (gray), t=0



(b) Favorable zone (blue) and $\mathcal{A}_i$'s operational zone (gray) at some t>0



(c) Mode Switching Automaton

Fig. 3: Aerial robots' zones and mode switching strategy

that either the entire human body or some part of the human's body is within the field-of-view of their on-board RGB sensor. To ensure this, the camera on the aerial robot needs to be pointed towards the human. We design the following yaw rate control for robot $\mathcal{A}_i$ that forces the RGB sensor to be always pointed toward the human:

$$\omega_{z,i} = \dot{\psi}_{i,d} - k(\psi_i - \psi_{i,d}) \tag{10}$$

where $\psi_{i,d} = \tan^{-1}(\frac{y_h - y_i}{x_h - x_i})$ is the desired yaw angle for the robot $\mathcal{A}_i$. Next, we describe a trajectory generation scheme for the aerial robots that uses head orientation feedback to move them to appropriate locations in the favorable zone $\mathcal{Z}_f$ to have better gaze estimate.

A. Trajectory Generation

To ensure that the robots look at the human or their face from an appropriate position, while ensuring they stay in their assigned zones $\mathcal{Z}_i$, we propose two different modes of operation : 1) Face tracking mode, and 2) Body tracking mode. These modes are discussed next.

1) Face Tracking Mode: The goal in the *face tracking* mode is to ensure that the aerial robot $\mathcal{A}_i$ is always within the favorable zone $\mathcal{Z}_f$ and stay as much in the front of the human as possible. This ensures proper human face detection to enable an accurate estimate of the head orientation and gaze. To ensure the aerial robot $\mathcal{A}_i$ is within the favorable zone $\mathcal{Z}_f$, and that the robot is as much in the front of the human as possible, we design the following potential function:

$$V_{f,i} = V_{p_i}(\mathbf{p}_h)(1 - O_i(\mathbf{r}_h)) \tag{11}$$

where $V_{p_i}(\mathbf{p}_h) = \ln\left(\frac{c_1 d_i' - c_2}{d_i' - c_0} + \frac{d_i' - c_0}{c_1 d_i' - c_2}\right)$, where d_i' is the distance between the human head and robot $\mathcal{A}_i$ in the x-y plane, $c_0 = \underline{d}^{\infty}$, $c_1 = \frac{\breve{d} - \underline{d}^{\infty}}{\bar{d}^{\infty} - \breve{d}} > 0$, $c_2 = c_1 \bar{d}^{\infty}$. The potential function $V_{p,i}$ approaches infinity when the planar distance d_i' approaches $\underline{d}^{\infty}$ from right or approaches $\bar{d}^{\infty}$ from left, and has a minima at the distance $d_i' = \breve{d}$. The parameter $\bar{d}^{\infty}$ is chosen based on the analysis provided in Section IV to be $\bar{d}^{\infty} = \bar{d}_i'$, and $\underline{d}^{\infty} = 1.5$ m to ensure aerial robots stay sufficiently far from the human for human's safety. We choose $c_1 = 1$ which yields $\breve{d} = \frac{\underline{d}^{\infty} + \bar{d}^{\infty}}{2} = 8.25$ m. Robot $\mathcal{A}_i$ aims to minimize this function when in *face tracking* mode.

2) Body Tracking Mode: When the favorable zone $\mathcal{Z}_f$ does not intersect with the operational zone $\mathcal{Z}_i$ of the aerial robot $\mathcal{A}_i$, the robot still needs to keep collecting information about the human. The goal in the *body tracking* mode is to ensure that the aerial robot is in the operational zone $\mathcal{Z}_i$ and the entire human body is visible to the on-board camera. These objectives are encoded in the following potential function for $\mathcal{A}_i$:

$$V_{b,i} = V_{p_i}(\mathbf{p}_h)\frac{\hat{\mathbf{z}}_{c,i}^T(\mathbf{p}_i - \mathbf{p}_h)}{d_i} \tag{12}$$

where $\hat{\mathbf{z}}_{c,i}$ is a fixed unit vector for robot $\mathcal{A}_i$ that forces $\mathcal{A}_i$ to stay in the center of the operational zone $\mathcal{Z}_i$ and is defined as:

$$\hat{\mathbf{z}}_{c,i} = [\cos(\psi_{zc,i})\cos(\theta_b),\ \cos(\psi_{zc,i})\cos(\theta_b),\ \sin(\theta_b)]^T \tag{13}$$

where $\psi_{zc,i} = \frac{\psi_{oz,i} + \bar{\psi}_{oz,i}}{2}$, and $\theta_b\ (\geq 0)$ is a constant elevation angle to ensure the robot stays below the height of the head to cover the entire body. Similar to *face tracking* mode, the goal for the robots is to minimize $V_{b,i}$ in the body tracking mode.

3) Height limits: To ensure the aerial robot $\mathcal{A}_i$ does not fly higher than the maximum height limit, we design the following barrier function:

$$V_{z,i} = -\ln(c_g + m_g d_i' + z_i) \tag{14}$$

The barrier function $V_{z,i}$ approaches infinity as z_i approaches the value $-(c_g + m_g d_i')$ from right.

4) Robot safety: While the robots move around the human, they should ensure they do not collide with

each other. To encode this requirement, we design the following Lyapunov-like barrier function:

$$V_{s,i}^{j} = -\ln(d_{ij}^2 - \rho_{safe}^2). \tag{15}$$

where $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$. The velocity correction to avoid collision between $\mathcal{A}_i$ and any other robot is defined as:

$$\mathbf{v}_{s,i} = \textstyle\sum_{j\in I} \sigma_{s,i}(d_{ij})\nabla_i(V_{s,i}^{j}) \tag{16}$$

where ∇_i is the gradient of the input argument with respect to the position $\mathbf{p}_i$, $\sigma_{s,i}(d_{ij})$ is a bump function [27] that activates the collision avoidance control $\mathbf{v}_{s,i}$ only in local neighborhood of the other agents and ensures the control velocity is continuous.

5) Robot Control Action and Mode Switching: The robot $\mathcal{A}_i$ applies the following gradient based control:

$$\mathbf{v}_i = \mathbf{v}_h + \sigma_{b,i}\nabla_i(V_{b,i}) + \sigma_{f,i}\nabla_i(V_{f,i}) + \sigma_{z,i}\nabla_i(V_{z,i}) + \mathbf{v}_{s,i} \tag{17}$$

where $\sigma_{f,i}$ and $\sigma_{b,i}$ are bump functions, defined later on, dictating which mode the robot $\mathcal{A}_i$ is operating in - *face* or *body* tracking mode, respectively (see Figure 3c). The function $\sigma_{z,i}$ is a bump function similar to $\sigma_{s,i}$ that activates the control to stay below the height limits in the neighborhood defined by (9). The bump function $\sigma_{f,i}$ is defined as

$$\sigma_{f,i} = \begin{cases} 1, & \text{if } \underline{\psi}_{oz,i} \le \hat{\psi}_h \le \bar{\psi}_{oz,i} \\ \sum_{k=0}^{3} G_{i,k}^{-}(\hat{\psi}_h)^k, & \text{if } \underline{\psi}_{oz,i} - \delta_\psi \le \hat{\psi}_h \le \underline{\psi}_{oz,i} \\ \sum_{k=0}^{3} G_{i,k}^{+}(\hat{\psi}_h)^k, & \text{if } \bar{\psi}_{oz,i} \le \hat{\psi}_h \le \bar{\psi}_{oz,i} + \delta_\psi \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $\hat{\psi}_h$ is an estimate of the yaw angle of the human head, δ_ψ provides a small angular zone on both ends of the operational zone of an aerial robot to allow smooth transition between the *face* and *body* tracking modes, $G_{i,k}^{-}$ and $G_{i,k}^{+}$ for all $k \in \{0,1,2,3\}$ are coefficients to ensure $\sigma_{f,i}$ is continuously differentiable and varies from 0 to 1 and from 1 to 0 in the respective transition zones defined by δ_ψ around operation zone of each robot. We further define $\sigma_{b,i} = 1 - \sigma_{f,i}$. These bump functions allow continuous velocity control during the transition between *body-tracking* and *face-tracking* mode. In summary, the controller in (17) ensures that robot $\mathcal{A}_i$ always stays inside its operational zone, stays safe from the human and the other robots, and either tracks only human body or the human face depending on operation mode.

VI. Results

We study the performance of the proposed algorithm using simulations in AirSim. The robots and a human are launched in an Unreal environment in a outdoor setting. The human moves around the environment and assesses that some tasks should have priority over other tasks. As the human looks at tasks, the aerial robots capture that information and communicate the preference to the ground robots. The ground robots then utilize a task allocation algorithm (Consensus-Based Bundle Algorithm (CBBA) [15], explained in the Appendix), to determine which ground robot goes to which task.

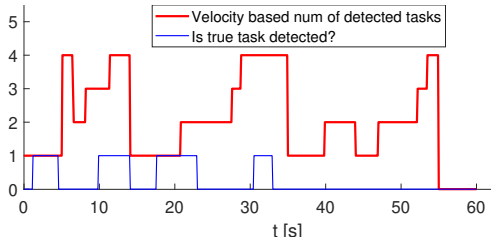


Fig. 4: Human velocity-based task identification

We consider a scenario where there are multiple objects on the ground (numbered Task 1 to Task 7), some of which need to be collected and stored in a nearby storage yard. The human walks around these tasks in sequence from Task 1 to Task 7 and gazes at these task locations for sufficient amount of time in order to indicate that these objects need to be picked and stored at specific gathering locations in the storage yard. The aerial robots use the algorithm described in Sections III-V to navigate around the human and gather the gaze information. This information is used in the CBBA so that each ground robot is assigned a task, reaches the assigned task location, picks up the object, and takes it to the storage yard to store it to a predefined location before going to its next task.

We consider three cases: 1) a simple human-velocity-based task identification, without considering any visual information about the human (baseline algorithm), 2) *No active face tracking*: only *body tracking* mode is active on all the robots, 3) *Active face tracking*: robots switch between the *body tracking* and the *face tracking* mode according to the switching strategy discussed in V-A.5. In Case 1), we create a 2D cone with vertex at the current position of the human and axis along the current direction of human velocity and with angular width of 25 degrees. All tasks inside this cone are considered to be tasks of interest to the human. In Figure 4, we show the number of tasks (red) that are identified by the human velocity-based approach and also show whether the true task is identified or not (true). As illustrated by the results, on average more than 3 tasks, in addition to the true task, were identified at any time instance. Although the true task is among the identified tasks most of the times, it is very difficult to determine which of the detected tasks is a true task (possible criteria are minimal distance, minimal angular offset, object properties, etc.).

For Case 2 and Case 3, we estimate the human gaze direction from the images captured by the robots and deploy the proposed algorithm to move aerial robots. We find the error between the true gaze direction $\mathbf{g}_h$, obtained from Unreal, and the estimated one $\hat{\mathbf{g}}_h$. We use the angle between the true gaze and the estimate as the metric for evaluation of the proposed scheme, and plot the angle of each case in Figure 5. The gaze error in case of *no active face tracking* is in general larger

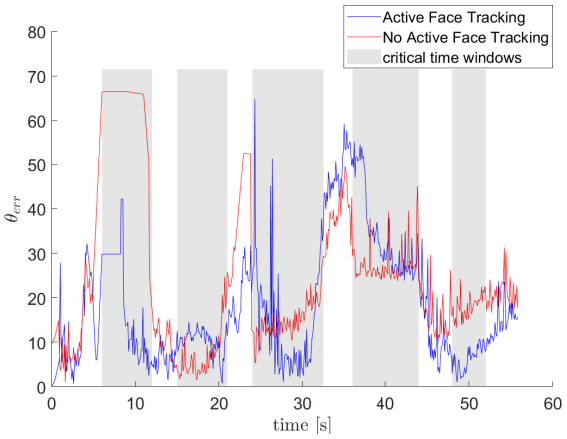


Fig. 5: Gaze error comparison for two scenarios

than that in the *active face tracking case*. The critical time windows during which the human actually starts looking towards the intended tasks are highlighted in gray. In some of the critical time windows, gaze error is similar or even better in *no active face tracking* case due to the fact that one of the robots is directly in front of the human in *no active face tracking* case. The gaze error in these instances majorly depends on the performance of the XGaze algorithm, which has its own gaze estimation error which gets reflected in Fig. 5. The videos of these simulations showing the motion of the human, aerial and ground robots; and the visual data available from the aerial robots are available at https://tinyurl.com/7h6zdjzf.

In Cases 2 and 3, whenever the human gazes at a particular object for a certain amount of time, that task is considered to be prioritized by the human to be executed. The ground robots then get assigned periodically to these tasks. The task identified based on the gaze information gathered by the aerial robots are highlighted by blue dotted circles in Figure 6. As many as three tasks are created in *active face tracking* mode. However, in the case of *no active face tracking*, only 2 tasks are created. This can be attributed to the lack of face detection and in turn that of gaze information during the segment when the human was walking from task 2 to task 3 (first gray zone in Figure 5) because no aerial robot was in the favorable zone in front of the human for proper gaze estimation. In most of instances, *no active face tracking* mode works equally comparable to *active face tracking* mode. This is incidental because, during these time intervals, the human face happens to be directed toward one of the body tracking robots.

It is also possible to get the same level of human head and gaze coverage using *body tracking* mode (no active face tracking) only, however, ensuring full coverage would require deployment of more aerial robots than when the robots were actively tracking the face.

Remark 1: We were not able to read the real-time position of the animated human model from Unreal environment in our AirSim python API, so we used the server's world clock time to determine the ground-truth time and human position in the environment. This led to some discrepancy between the robots' time and human's time, which may have affected the numbers shown above. However, performing the simulation for several times yielded similar results, which were all consistent with the overall qualitative behavior of the algorithm as discussed above.

VII. Conclusions and Ongoing Work

In this paper, we considered the development of a multi-robot system that operates near humans in their working environment, while visually observing them at work and offering assistance when needed. We developed and evaluated a collaborative control algorithm for a team of aerial robots that actively tracks human head and gaze in order to improve gaze estimation quality, and reliably infer the location of the human's target objects by following the human's gaze direction. The algorithm is tested in AirSim and evaluated against a baseline algorithm. Our simulations showed that actively tracking the human by moving aerial robots around the human is more effective in terms of the efficiency in understanding the human intentions, as well as in terms of the number of aerial robots deployed. We demonstrate the superiority of the suggested approach in comparison to other approaches, including a static or dynamic visual tracking of human-body only, and a naive velocity-based intention identification. Future work will focus on experimental implementation and evaluation of the proposed multi-robot coordination for gaze following and human-intent inference.

ACKNOWLEDGMENT

The research was supported by D. Dan and Betty Kahn Michigan-Israel Partnership. D.H. was supported by the Robin Neustein Artificial Intelligence fellowship.

Appendix

A set of ground robots are assigned the tasks using a Consensus Based Bundle Algorithm (CBBA) [15], which is a decentralized task selection algorithm that utilizes a market-based decision strategy. The CBBA algorithm has two phases. Phase 1 includes bundle construction. Each agent carries two types of lists of tasks: the bundle list and the path list. Tasks in the bundle are ordered based on time added, while tasks in the path list are ordered based on location in the assignment. When a task is added to the bundle list, it receives a score improvement, where the total reward value is based on an agent performing tasks along a path. Detailed description is given in [15].

In this paper, we modify the cost function for creating the bundle and path list by considering when the human's gaze intersects with a target location (task). Preference is incremented by a fixed number while the human is looking at a task. The CBBA algorithm generates a new path list at each iteration based on the new

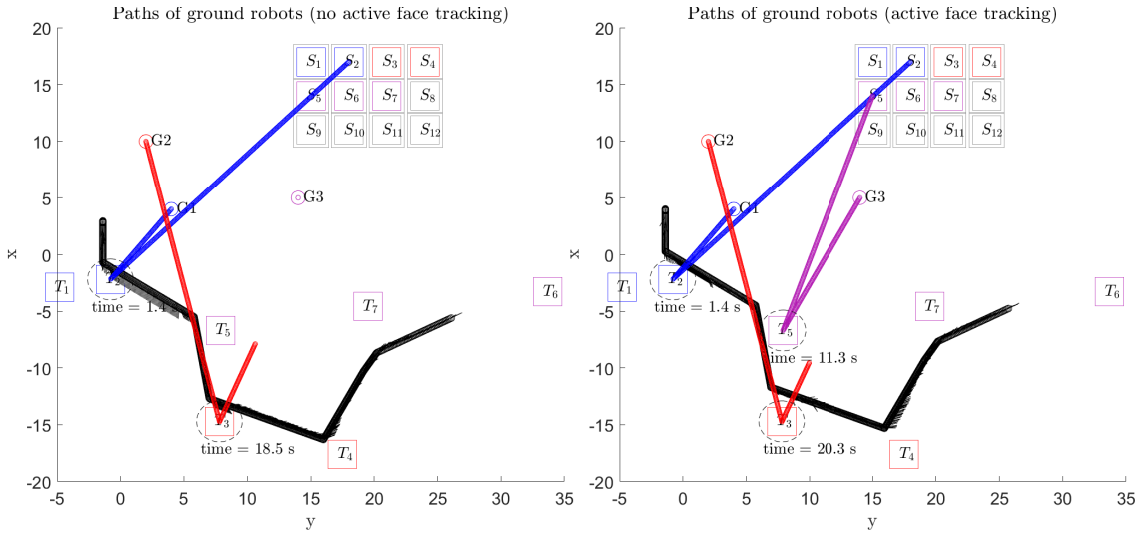


Fig. 6: Paths followed by the human and the ground robots

human-preference values. Once an agent arrives at and completes a task, the task is removed from the task list.

References

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," Autonomous Robots, vol. 42, no. 5, pp. 957–975, 2018.

[2] R. Poppe, "Vision-based human motion analysis: An overview," Computer vision and image understanding, vol. 108, no. 1-2, pp. 4–18, 2007.

[3] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," 2010.

[4] X. Wang, K. Liu, and X. Qian, "A survey on gaze estimation," in 2015 10th Int. Conference on Intelligent Systems and Knowledge Engineering (ISKE). IEEE, 2015, pp. 260–267.

[5] L. Zhang and R. Vaughan, "Optimal robot selection by gaze direction in multi-human multi-robot interaction," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016, pp. 5077–5083.

[6] X. Wang, J. Zhang, H. Zhang, S. Zhao, and H. Liu, "Vision-based gaze estimation: a review," IEEE Transactions on Cognitive and Developmental Systems, 2021.

[7] N. F. Duarte, M. Rakovic, J. S. Marques, J. Santos-Victor, L. Leal-Taixe, and S. Roth, "Action alignment from gaze cues in human-human and human-robot interaction." in ECCV Workshops (3), 2018, pp. 197–212.

[8] K. Fujii, G. Gras, A. Salerno, and G.-Z. Yang, "Gaze gesture based human robot interaction for laparoscopic surgery," Medical image analysis, vol. 44, pp. 196–214, 2018.

[9] L. Paletta, M. Pszeida, H. Ganster, F. Fuhrmann, W. Weiss, S. Ladstätter, A. Dini, S. Murg, H. Mayer, I. Brijacak et al., "Gaze-based human factors measurements for the evaluation of intuitive human-robot collaboration in real-time," in 2019 24th IEEE Int. Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2019, pp. 1528–1531.

[10] L. Yuan, C. Reardon, G. Warnell, and G. Loianno, "Human gaze-driven spatial tasking of an autonomous MAV," IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1343–1350, 2019.

[11] M. Pavliv, F. Schiano, C. Reardon, D. Floreano, and G. Loianno, "Tracking and relative localization of drone swarms with a vision-based headset," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1455–1462, 2021.

[12] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," in European Conference on Computer Vision. Springer, 2020, pp. 365–381.

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," The international journal of robotics research, vol. 5, no. 1, pp. 90–98, 1986.

[14] Y. Shin and E. Kim, "Hybrid path planning using positioning risk and artificial potential fields," Aerospace Science and Technology, vol. 112, p. 106640, 2021.

[15] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," IEEE Transactions on Robotics, vol. 25, no. 4, pp. 912–926, 2009.

[16] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in Field and Service Robotics, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, 2012.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[19] N. Krüger, M. Pötzsch, and C. von der Malsburg, "Determination of face position and pose with a learned representation based on labelled graphs," Image and vision computing, vol. 15, no. 8, pp. 665–673, 1997.

[20] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pp. 607–626, 2008.

[21] M. J. Marin-Jimenez, A. Zisserman, M. Eichner, and V. Ferrari, "Detecting people looking at each other in videos," International Journal of Computer Vision, vol. 106, no. 3, pp. 282–296, 2014.

[22] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2074–2083.

[23] D. E. King, "Dlib-ml: A machine learning toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755–1758, 2009.

[24] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3d solution," in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 146–155.

[25] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba, "Where are they looking?" Advances in neural information processing systems, vol. 28, 2015.

[26] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4511–4520.

[27] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed coordination control for multi-robot networks using lyapunov-like barrier functions," IEEE Transactions on Automatic Control, vol. 61, no. 3, pp. 617–632, 2015.