HRI in the Sky: Creating and Commanding Teams of UAVs with a Vision-mediated Gestural Interface

Valiallah (Mani) Monajjemi, Jens Wawerla, Richard Vaughan and Greg Mori*

Abstract— **Extending our previous work in real-time vision-based Human Robot Interaction (HRI) with multi-robot systems, we present the first example of creating, modifying and commanding teams of UAVs by an uninstrumented human. To create a team the user focuses attention on an individual robot by simply looking at it, then adds or removes it from the current team with a motion-based hand gesture. Another gesture commands the entire team to begin task execution. Robots communicate among themselves by wireless network to ensure that no more than one robot is focused, and so that the whole team agrees that it has been commanded. Since robots can be added and removed from the team, the system is robust to incorrect additions. A series of trials with two and three very low-cost UAVs and off-board processing demonstrates the practicality of our approach.**

I. Introduction

Selecting and commanding individual robots in a multi-robot system can be a challenge: interactions typically occur over a conventional on-screen human-computer interface (e.g. [1]), or specialized remote control (e.g. [2]). Humans, however, can easily select and command one another in groups using only eye contact and gestures. We are working on non-verbal communication methods for human-robot interactions. In particular we avoid the need for the human to be instrumented in any way, and all interaction is mediated by the robot's on-board sensing and actuation.

In this paper we extend our previous work [3] using face engagement to select a particular robot from a group of robots. In our previous system, once selected, a single robot engaged in one-on-one interaction with the user. In this paper we compose a multi-robot team from the population of robots by adding or removing the currently selected robot, then command the whole team at once. In our previous paper we used wheeled mobile robots which were stationary for the human-robot interactions. In this paper we use flying quadrotor robots which are continuously moving. The constant movement of cameras attached to flying robots make the problem of vision mediated human robot interaction much more challenging.

The contributions of this work are: (i) the first demonstration of HRI control of a flying robot by an uninstrumented human using only passive computer vision; (ii) the first demonstration of dynamically creating and modifying robot teams by an uninstrumented human; and (iii) the first demonstration of focusing attention on a flying robot by face-engagement.

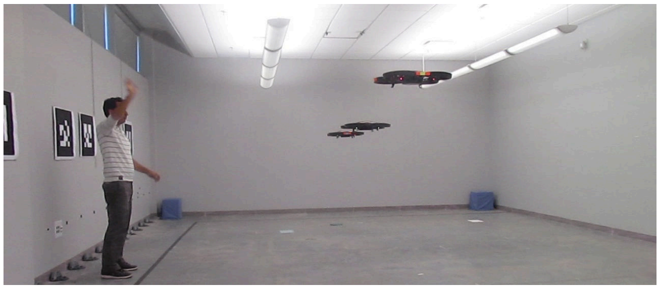

Fig. 1. An uninstrumented person creates and commands a team of three UAVs using face-engagement and hand gestures

II. Background

Throughout this work, we will use the term *face engagement*, as coined by Goffman, to describe the process in which people use eye contact, gaze and facial gestures to interact with or engage each other [4].

A. Uninstrumented Human Robot Interaction

Researchers have argued that exploiting stereotypical communication cues (without instrumentation) can achieve natural human-robot interactions [5]. Gaze and body movements (gestures) are two such communication cues.

There is a large literature on gaze tracking techniques; Morimoto and Mimica provide an in-depth survey [6]. Applications of gaze trackers can be found in fields ranging from psychology to marketing to computing science; many interesting examples are given in the survey provided by Duchowski [7].

In an experiment by Mutlu et al. [8], gaze is used to regulate conversations between, a humanoid robot, and two human participants. The study showed that (among other things) gaze was an effective tool for yielding speaking turns and reinforcing conversation roles. Kuno et al. [9] present a museum tour-guide that only responds when directly looked at. A telephoto lens is used to capture a high quality image; the robot then estimates if the user is looking at it by detecting if the nostrils are centered between the eyes. Our previous work showed that this basic method can be extended to select individual robots from a population by using explicit wireless communication between robots to perform a distributed election algorithm to unambiguously decide which robot (if any) was being looked at directly [3]. Since the election is completed in a few tens of milliseconds and is essentially imperceptible to the user, the users experience is simply that as you look from robot to robot, the selected

* School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. {mmonajje, jwawerla, vaughan, mori}@sfu.ca

robot is always “the one I am looking at right now”. Below we show that this method is also effective for flying robots.

There is a vast computer vision literature on gesture recognition: Mitra and Acharya [10] provide a survey. Several gesture-based robot interfaces exist; systems may use static gestures – where the user holds a certain pose or configuration – or dynamic gestures – where the user performs a combination of actions. Waldheer at al. use both static and motion-based gestures to control a trash-collecting robot [11]. Earlier work by Kortenkamp et al. presents a mobile robot that uses an active vision system to recognize static gestures by building a skeleton model of the human operator; a vector of the human’s arm is used to direct the robot to a particular point [12]. Giusti et al. [13] demonstrated how a swarm of mobile robots can cooperatively detect a static human gesture and act upon it.

We use simple motion-based gestures to issue commands to robots once they have been selected using face-engagement.

B. Robot Selection And Task Delegation

There is little work on human-robot interfaces for multi-robot systems. Examples can be broken up into two general cases:

1) World-Embodied Interactions: World-embodied interactions occur directly between the human and robot, through either mechanical or sensor-mediated interfaces. Key advantages of this approach compared to a conventional GUI include the possibility for users to walk freely among the robots rather than being tied to an operator station. Also since robots observe humans directly using their on-board sensing, they may not need to localize themselves in a shared coordinate frame. Examples include work by Payton that uses an omni-directional IR LED to broadcast messages to all robots, and a narrow, directional IR LED to select and command individual robots [2], work by Naghsh et al. [14] who present a similar system designed for firefighters, but do not discuss selecting individual robots , and work by Zhao et al. [15] which proposes the user leaves fiducial-based “notes” (e.g. “vacuum the floor” or “mop the floor”) for the robots at work site locations. Xue et al. [16] introduced a fiducial design for imperfect visibility conditions and combined them with user-centric gestures.

2) Traditional Human-Computer Interfaces: Rather than interacting directly with robots, a traditional human-computer interface is used to represent the spatial configuration of the robots and allow the user to remotely interact with the robots. Examples of human-robot interactions which occur through a traditional interface include work by McLurkin et al. [1] that presents an overhead-view of the swarm in a traditional point and click GUI named “SwarmCraft”, and work by Kato that displays an overhead live video feed of the system on an interactive multi-touch computer table, which users can control the robots’ paths by drawing a vector field over top of the world [17]. Similar to Zhao et. al’s fiducial-based notes [15], Kolling et al. [18] designed a user interface that allows the operator to place virtual beacons in a simulated robot environment.

C. Human Robot Interaction with UAVs

Traditional human computer interfaces have been used extensively to design control interfaces for single [19]–[21] and multiple [22] UAVs. Uninstrumented interfaces have also been used to interact with UAVs. Song et al. [23] describes a method for recognizing aircraft handling signals from depth data, and tested their method on a database of videos collected from a stationary (non-airborne) camera. Lichtenstern et al. [24] describe a prototype system in which gestures directed at one UAV carrying a Kinect (active RGB-D) sensor can be used to control other UAVs. Jones et al. [25] performed a user study to investigate how different modalities can be used to control a swarm of simulated UAVs in a virtual reality environment. Naseer et al. [26] developed an autonomous system that enables a single quadrocopter to follow a human and respond to hand gestures using active RGB-D sensor with vision-based ego-motion cancellation.

Our work is different from the aforementioned works due to our use of vision-based gestures (obtained from a passive monocular camera) to select and command a team of airborne UAVs. Now that affordable UAVs are available we expect this area to grow rapidly.

III. Method

To demonstrate our approach, we use a group of unmodified AR-Drone 2.0 quadrocopters[1]. These inexpensive aircraft have a built-in attitude controller and a forward-facing 720p HD camera. Video from the camera and flight control data are streamed via 802.11 wireless network to a control computer. A practical challenge when using this setup is that all user software is run externally and is therefore subject to large network delays: we observe around 200 milliseconds end-to-end latency. Engel et al. [27] have shown that it is possible to explicitly model the communication delay and use monocular Simultaneous Localization and Mapping (SLAM) to accurately navigate a single quadrocopter. Another successful position controller is presented by Krajník et al. [28]. They determined the drone’s transfer function and implemented a PID controller that would hover the drone over a mobile target, tracked by the downward facing camera. We use only the forward facing camera for HRI and localization, since the platform does not permit simultaneous streaming from both cameras.

Next we describe our approach, with an overview shown in Fig. 2.

A. Position Estimate and Control

While the AR-Drone 2.0 is capable of generating 720p video streams, we use a lower resolution to save wireless channel bandwidth and allow us to use multiple robots. We experimented with two different 3D pose estimation methods for the robots: fiducial based and ambient feature based. The fiducial based method uses the ALVAR library [29] to track

[1] `http://ardrone2.parrot.com/`

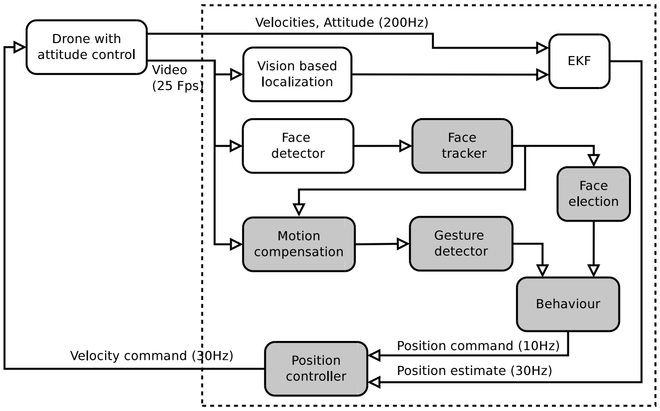

Fig. 2. System overview, the dashed box (right) wraps the components that run on a laptop, the remainder (left) runs on-board the aircraft. Components in gray (lower right) are custom developed for this work, while third party modules with small adaptations are marked in white.

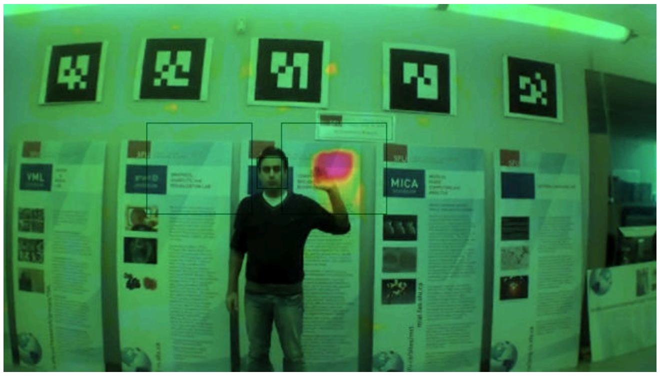

Fig. 3. Face detection is used to locate the user, and to select the currently focused robot. Hand gestures change the state of the focused robot. This image is from the flying robot's point of view. The gesture detection regions are marked by a rectangle. (The stabilized optical flow magnitude's heat map is blended into the image.)

the drone's position $(x, y, z, \phi, \theta, \psi)^T$ relative to fiducials mounted at known locations in the environment. Here x, y, z is the 3D location in the world frame and ϕ, θ, ψ are roll, pitch and yaw (heading), respectively. The feature-based method employs the Parallel Tracking and Mapping (PTAM) monocular SLAM system [30] to estimate each robot's pose. We use an Extended Kalman Filter (EKF) to fuse the vision based position estimate with inertial measurements from the drone's flight control computer to improve the accuracy of the pose estimate.

When robots use the fiducial based method, they are localized in the global coordinate frame, which makes the multi-robot formation control straightforward. However, this method is sensitive to fiducial occlusions. The feature-based method on the other hand is more robust to occlusions. However, the coordinate frame and scaling of pose estimates are not defined with respect to the world and depend on the PTAM initialization phase. Our system uses the method introduced in [27] to perform scale estimation using EKF. In our system, all robots use the same recorded video of the environment for PTAM initialization, and thus they all agree on the initial coordinate frame.

To control each drone, the position estimate and 4-DOF target position $(x_T, y_T, z_T, \psi_T)^T$ are fed into four independent PID controllers, one for each directly controllable degree of freedom. The control output is then sent via the wireless network to the drone. In practice we find that this approach works well as long as there is sufficient distance ($> 3m$) between any two aircraft. When drones are too close together, turbulence from the down draft causes the drones to pitch and roll rapidly in an attempt to maintain their position, and the camera can not be kept on-target for HRI. This fast movement cannot sufficiently be tracked by our position controller because of the network delay. We avoid this issue by enforcing a minimum distance of $3m$ between aircraft.

B. Face Detection and Tracking

To locate and track faces in the video stream, we use the OpenCV [31] implementation of the Viola-Jones [32] face detector. Because of the often rapid ego-motion of the airborne camera we might lose a detected face or detect several false positives. We address this problem by using a Kalman Filter to smooth face position estimates. We use a nearest neighbor data association strategy to determine which detected face to use as the measurement input, using a Mahalanobis distance derived from the estimated covariance of candidate faces.

Information about the tracked face is used in two subsequent modules: first to partially cancel image flow due to ego-motion as described in the next section, and second to determine if the user is engaging in an interaction with the robot. Our HRI attention-focusing strategy is to engage one robot at a time out of the group by simply looking at it. Subsequent commands are addressed to the engaged robot. The challenge for the robots is to determine which robot is currently being looked at, as the user's face might be visible to several robots at the same time. We use a mechanism developed and successfully used earlier by our group [3]. The face detector is trained on frontal faces only, and we observe that the largest number of candidate face detections occur when the face is looking directly at the camera. Since the face detector is insensitive to small changes in scale or position, multiple candidate detections are often clustered around faces. We use the number of candidate detections in each cluster as a score to assess the quality of the detected face. To determine which robot sees the most frontal face the robots perform a distributed election, each proposing their currently observed face score. If no robot has a score above a threshold, no robot is engaged, otherwise the robot with the highest score is the one being engaged by the human. Only the currently engaged robot will watch for gestural commands.

C. Motion Cancellation and Gesture Recognition

The system uses the magnitude of optical flow in fixed regions around the user's face to detect hand-wave gestures. In order to have reliable optical flow information, motion from sources other than user's hand movement in the video

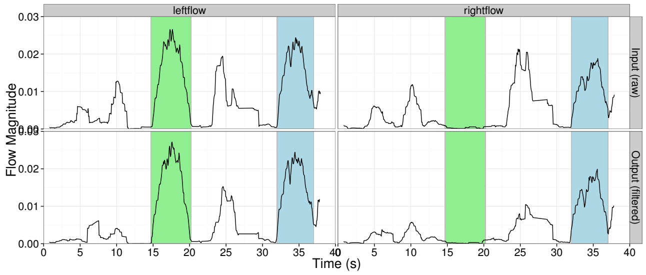

Fig. 4. Optical flow in the left and the right hand zone; the top graph shows the unfiltered optical flow and the bottom graph shows the output of our multi-stage filter. Sections marked in green (left) correspond to left hand gesture, periods of both hands gesture are colored blue (right).

stream should be filtered out. We have to deal with three sources of motion in our video stream. The first is the motion of the camera caused by the motion of the aircraft stabilizing its attitude and controlling its position. The second is caused by user movements other than gesturing, and the third is a result of the hand gestures used to command the vehicle. The objective is to cancel the first two while not damping the gesture motion.

For motion cancellation and gesture recognition we define three zones in the image. The face zone is a bounding box around the face currently being tracked. The left and right hand zones are rectangles to the left and right of the face box respectively as shown in Fig. 3. The size of the left and right zones is proportional to the size of the face zone. The hand zones are cropped if any of their corners exceed the image boundaries. This will happen when the human face is towards the edges of the image.

In a first step we mask all pixels in the hand zones to preserve the optical flow caused by waving the hands. We then calculate optical flow in the remainder of the image using the OpenCV [31] implementation of Franebäck's algorithm [33]. The median of this optical flow is an approximation of the ego-motion of the camera, which we can now remove from the original image.

Next, using the camera-motion-reduced image, we estimate the motion of the user by computing the median of the optical flow in just the face zone. The assumption is that motion of the face is a reasonable proxy overall non-gesture body motion. By removing the estimated user motion from the image we are left with an image that contains mainly the flow resulting from the gestures. The process is illustrated in Fig. 4.

In the last step we average the magnitude of optical flow within the hand zones. For robustness to transient flow, the resulting signal is passed through a median filter with a window size of 15 frames. By thresholding the result we can detect left and right hand waving. This gives us a total of 4 states: no wave, left wave, right wave and two-hand wave. These gestures are then used by the behavioral module to command the aircraft.

D. Commanding the Vehicle

The user commands a robot by first engaging it (by looking at it) and then giving it one of the three gestures. A right hand wave means join the group. A robot that is part of the group increases its hover altitude by 0.2m. A left hand wave is the command to leave the group, consequently the aircraft returns to the original altitude. Waving both hands is the signal for the entire group to execute a mission. Note that only one robot has to be given the command to execute the mission; it will communicate this instruction to the others over the network and the group acts as one. In our demonstration the "mission" is either to land or perform a complete roll (flip) in place. These simple missions are a placeholder for a real mission such as search, patrol, mapping, etc. The robots also change the color and blinking frequency of their built-in LEDs to report their current state (being engaged or selected as part of the group) to the user. Informally, we found this direct feedback helps the user in the interaction process.

The flowchart of the controller is shown in Fig. 5. We trigger take-off manually. Each aircraft, once airborne, autonomously flies to its predefined target location and tries to detect faces. If a face is detected as described above the position controller tracks the face by steering the nose of the aircraft in the direction of the face. This is to ensure that the face is always in the middle of the image. This is not only a feedback mechanism to the user, but also keeps the hand zones from being cropped. Next, the face scores are communicated to all robots by wireless network. If a robot wins the face score election, it considers itself engaged by the user and accepts hand gestures. Left or right hand gestures set or clear a "belong to group" flag. If the *execute command* gesture is detected, the command is passed on to all other aircraft via the wireless network. An aircraft receiving the execute command and belonging to the group will now execute the mission, i.e. land. The remaining aircraft stay airborne and wait for a user engagement.

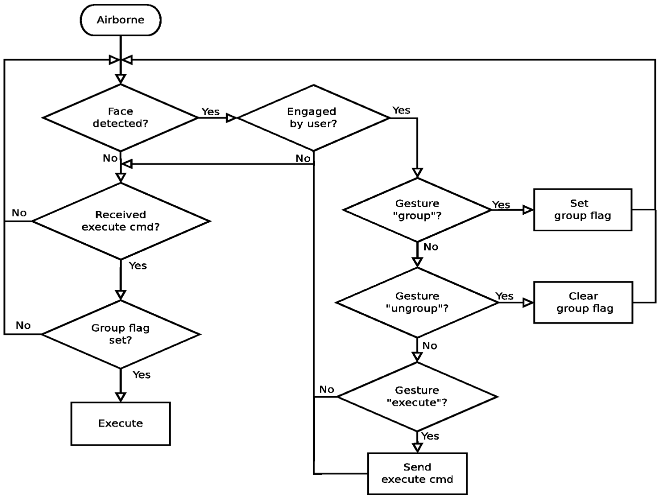

Fig. 5. Flowchart outlining the decision tree for the robot's behaviour.

TABLE I

Result summary for three robot experiment, S_i, D_i and C_i mean issue the Select, Deselect and Command gesture to the ith robot. Unintended outcomes are marked by overstrikes.

Trial	Scenario	Gesture	Face	Success
1	S_1 S_2 C_1	3/3	3/3	Yes
2	S_1 S_2 C_3 C_2	4/4	4/4	Yes
3	S_1 S_2 S_3 C_3	4/4	4/4	Yes
4	S_1 S_2 D_1 C_2	4/4	4/4	Yes
5	S_2 S_3 ~~D_2~~ ~~C_2~~ ~~C_3~~	2/5	5/5	No
6	S_2 S_3 S_1 D_2 C_3	5/5	5/5	Yes
7	S_3 S_2 S_1 D_2 C_3	5/5	5/5	Yes
8	S_2 S_1 S_3 ~~D_3~~ C_2	4/5	5/5	No
9	S_2 S_3 C_1 S_1 C_1	5/5	5/5	Yes
10	S_1 S_2 S_3 D_1 D_2 C_3	6/6	6/6	Yes
11	S_1 S_2 D_1 ~~D_3~~ C_2 S_1 C_1	6/7	7/7	No
12	S_1 S_3 D_1 ~~S_2~~ C_3 S_1 C_1	6/7	7/7	No
13	S_3 S_2 S_1 D_1 D_2 D_3 C_2	7/7	7/7	Yes
14	S_3 S_2 S_1 D_2 D_3 C_2 C_1	7/7	7/7	Yes
15	S_2 S_3 ~~D_2~~ S_1 D_3 D_1 S_3 C_3	7/8	8/8	No
Total		75/82	82/82	10/15

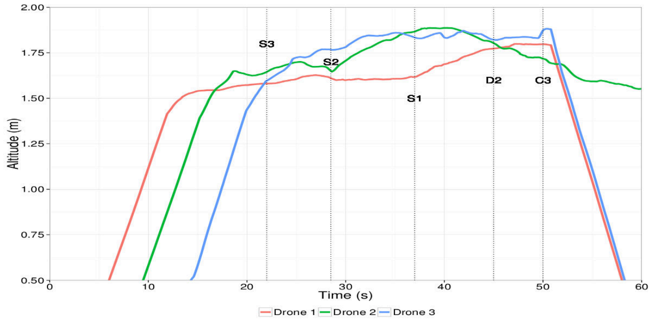

Fig. 6. Plot of smoothed robot altitude over time during trial #7 (Table I). Dotted vertical lines show the time that a specific gesture was performed. Select (S_i) adds robot i to the team, Deselect (D_i) removes robot i from the team. Team members hover 0.2m higher than non-team members. The Execute command (C_i) makes the team land.

IV. Demonstration

To demonstrate our system, we performed two sets of trials with a group of flying robots and a human. All trials were performed by one expert user. The arena is a 8 × 10 × 3m indoor lab environment clear of any static obstacles, shown in Fig. 1. At startup, each robot is placed at a pre-defined position on the ground. During each trial, the robots take-off after receiving an external signal, then fly to their pre-defined target poses $(x_T, y_T, z_T, \psi_T)^T$. The main difference between two sets of experiments are the position estimation method used for each experiment and the number of participating robots.

A. Three-Robot Experiment with Marker-Based Localization

In our first experiment, we used the fiducial based position estimation method as described in section III-A. Six unique 50×50cm ALVAR 2D tags were mounted on the wall behind the user as input to the ALVAR localization system. Due to low accuracy of heading estimates when the robots are looking at the fiducials with steep angles, initial poses for robots were set such that they look directly towards the fiducials. This led to a linear initial formation as shown in Figure 1. As a result, the human usually needs to walk along the wall into a robot's field of view first to get its attention. Once a face is seen by a robot, it yaws to track the face as described in section III-B.

TABLE II

Result summary for two robot experiment

Trial	Scenario	Gesture	Face	Success
1	S_1 S_2 C_1	3/3	3/3	Yes
2	S_1 S_2 C_2	3/3	3/3	Yes
3	S_1 S_2 D_1 C_2	4/4	4/4	Yes
4	S_1 S_2 D_2 ~~C_2~~ C_1	4/5	5/5	No
5	S_1 D_1 S_2 C_2	4/4	4/4	Yes
6	S_1 D_1 D_2 S_2 ~~C_1~~	4/5	5/5	No
7	S_2 S_1 D_2 D_1 S_2 C_2	6/6	5/5	Yes
8	S_1 S_2 D_2 S_2 C_2	5/5	5/5	Yes
9	S_1 S_2 S_1 C_2	4/4	4/4	Yes
10	S_1 S_2 D_2 D_2 S_2 C_1	6/6	6/6	Yes
Total		43/45	45/45	8/10

Fifteen trials with a total of 82 scripted interactions were executed. Table I summarizes the results. Robots were indexed from 1 to 3. In the table, the Scenario column contains a list of the interactions attempted by the user. S_i, D_i and C_i mean issue the Select (add to team), Deselect (remove from team) and Command (execute mission) gesture to the ith robot, respectively. Unintended outcomes are marked by overstrikes. A trial with any unintended outcome is deemed to be unsuccessful. The ratio of successful to overall trials was $10/15$. The success rate of individual interactions was $75/82$.

To summarize the robot system behavior, we recorded each robot's altitude for the length of the trial. Figure 6 shows such a graph for experiment number 7. The script was to select robot 3, select robot 2, select robot 1, deselect robot 2, then command robot 3 to land. The plot shows the altitude of robot 3 increasing at around 25 seconds, followed by robot 2 at around 30 seconds and 1 at 40 seconds, as each joins the team. The altitude of robot 2 decreases at around 45 seconds as it leaves the team. Robots 1 and 3 land at 60 seconds, while robot 2 remains hovering, as required by the trial script.

B. Two-Robot Experiment with Feature-Based Localization

In a second set of experiments we used the monoSLAM pose estimation method. The main motivation was to let the robots create a formation in which they initially look at the same spot in the room. We could not arrange this with the ALVAR-based system as the robots needed to face directly towards a fiducial to maintain a stable hover. With the PTAM monoSLAM method, this restriction is lifted and the user can stand on one spot and just look from robot to robot without moving in an out of the robots' field of view. The other benefit of this method is that there is no need to instrument the environment with fiducial markers.

This system has its own limitations though. The PTAM system is not able to track the position of the robot well when the camera motion is mainly rotational. This situation

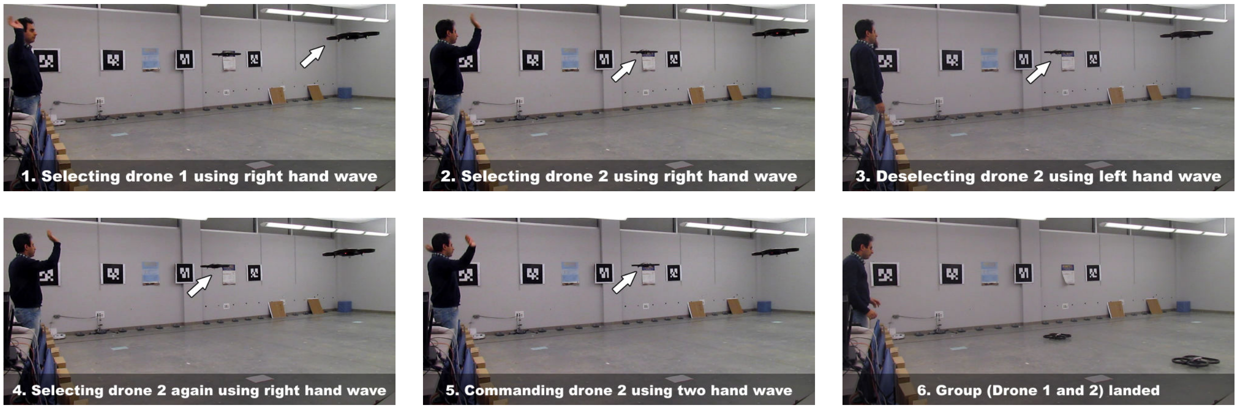
1. Selecting drone 1 using right hand wave
2. Selecting drone 2 using right hand wave
3. Deselecting drone 2 using left hand wave
4. Selecting drone 2 again using right hand wave
5. Commanding drone 2 using two hand wave
6. Group (Drone 1 and 2) landed

Fig. 7. Snapshots from a two robot experiment, in which a user is commanding two quadrocopters (Table II).

happens when the robot is tracking the human's face while hovering close to its target position. We found empirically that to avoid this situation, the change in heading of the robot should be small while performing stable hovering and face tracking. This means that, as the heading angle of the robot with respect to human increases, the distance between human and the robot should increase. This new constraint, in addition to the minimum distance constraint discussed in III-A, meant that we only had space for a two-robot experiment in our lab.

We performed a total number of 10 scripted trials with two drones. Table II summarizes the results. The ratio of successful to unsuccessful trials was $8/10$. The success rate of individual interactions was $43/45$. Figure 7 shows snapshots of trial number 8.

C. Discussion

In all trials the face engagement subsystem was successful: the robots could successfully detect and track the user's face while running the distributed leader election algorithm. We note informally that this capability combined with the LED and altitude feedback made a comfortable and natural-feeling method of interaction with the robots. The gesture recognition subsystem however had a total of 9 failures, 7 cases of false recognition and 2 cases of failed recognition. Examining the data, we found that false negative and incorrect recognitions occur when the motion cancellation happens to cancel a legitimate hand motion. The false recognition can also occur when the motion cancellation does not filter out all non-relevant motions.

The position control subsystem also had some failures when the marker based pose estimate of a robot became inaccurate either due to full occlusion of localization tags by the user's body or very fast human movements during an interaction. Although the robots could recover from these errors, their short-term instability forced the human to wait. After a few practice trials, the user learned to move his body so as to avoid these problems. While our ultimate goal is to design systems where such user adaptation is not necessary, we observe informally that a bit of user training can lead to a useful improvement in the performance of the current system. The occlusion was not a problem when using feature based pose estimates, however PTAM recovery after initialization from the pre-recorded video sometimes could take up to 15 seconds.

Most of the development time on this project was spent on localization and position control of these limited, low-cost UAVs. Our goal is to extend this system to be used in outdoor environments. In such cases, a GPS-based localization method can also be utilized to improve the quality of pose estimation and control.

V. Conclusion and future work

We presented a computer vision-mediated human-robot interface whereby an uninstrumented user can create, modify and command a team of robots from a population of autonomous individuals in a multi-robot system. The user selects an individual as the current focus of attention by simply looking at it. The focused robot can be added/removed from the team by waving the right/left hand. The whole team is dispatched to a mission by waving both hands.

We demonstrated the effectiveness of this method using a system of low-cost quadrotor robots with on-board attitude control and off-board computer vision-based 4-DOF position control. In a series of trials the robots achieved better than 90% correct execution of the user's intentions and 76% correct execution of trial interaction scripts.

A proper user-study with a naive participants would be required to justify a formal claim that this system is "intuitive" or better than any other method. We do not make this claim, but note informally that selecting a robot by looking at it is really fun, and even in our proof-of-concept implementation it is responsive and feels easy and natural.

We used a very small set of discrete gestures. The gestures set could be extended to allow a user to point to some arbitrary place in the environment, and have the robots fly to that location. This has been done for a single robot system (e.g. [12], [34]); however, an interesting extension would be to coordinate multiple robots to cooperatively estimate the vector given the system's ability to simultaneously capture images of the user from multiple angles (in the spirit of [13]).

The most urgent direction is to move outdoors. We aim to soon have robots flying over large distances doing useful tasks coordinated by the human user on the ground.

VI. Acknowledgments

This work was supported by NSERC in Canada. Thanks to the ROS community for software and support.

References

[1] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, *et al.*, "Speaking swarmish: Human-Robot interface design for large swarms of autonomous mobile robots," in *Proc. of the AAAI Spring Symp.*, 2006.

[2] M. Daily, Y. Cho, K. Martin, and D. Payton, "World embedded interfaces for human-robot interaction," in *System Sciences, 2003. Proc. of the 36th Annual Hawaii Int. Conf. on*, 2003, p. 6.

[3] A. Couture-Beil, R. T. Vaughan, and G. Mori, "Selecting and commanding individual robots in a vision-based multi-robot system," in *Proc. of Canadian Conf. on Computer and Robot Vision*, May 2010, pp. 159–166.

[4] E. Goffman, *Behavior in Public Places: Notes on the Social Organization of Gatherings*. Free Press, September 1966.

[5] B. R. Duffy, "Anthropomorphism and the social robot," *Robotics and Autonomous Systems*, vol. 42, no. 3–4, pp. 177–190, 2003.

[6] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[7] A. T. Duchowski, "A breadth-first survey of eye-tracking applications." *Behavior Research Methods, Instruments, and Computers*, vol. 34, no. 4, pp. 455–470, 2002.

[8] B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita, "Footing in human-robot conversations: how robots might shape participant roles using gaze cues," in *Proc. of Int. Conf. on Human Robot Interaction (HRI)*, 2009, pp. 61–68.

[9] Y. Kuno, M. Kawashima, K. Yamazaki, and A. Yamazaki, "Importance of vision in human-robot communication understanding speech using robot vision and demonstrating proper actions to human vision," in *Intelligent Environments*, ser. Advanced Information and Knowledge Processing. Springer London, 2009, pp. 183–202.

[10] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews*, vol. 37, no. 3, pp. 311–324, May 2007.

[11] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, pp. 151–173, 2000.

[12] D. Kortenkamp, E. Huber, R. P. Bonasso, and M. Inc, "Recognizing and interpreting gestures on a mobile robot," in *Proc. of the Nat. Conf. on Artificial Intelligence*, 1996, pp. 915–921.

[13] A. Giusti, J. Nagi, L. M. Gambardella, S. Bonardi, and G. A. Di Caro, "Human-swarm interaction through distributed cooperative gesture recognition," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE Int. Conf. on*. IEEE, 2012, pp. 401–401.

[14] A. M. Naghsh, J. Gancet, A. Tanoto, and C. Roast, "Analysis and design of human-robot swarm interaction in firefighting," in *IEEE Symp. on Robot and Human Interactive Communication (RO-MAN)*, 2008, pp. 255–260.

[15] S. Zhao, K. Nakamura, K. Ishii, and T. Igarashi, "Magic cards: a paper tag interface for implicit robot control," in *Proc. of Int. Conf. on Human Factors in Computing Systems (CHI)*, 2009, pp. 173–182.

[16] A. Xu, G. Dudek, and J. Sattar, "A natural gesture interface for operating robotic systems," in *Proc. of Int. Conf. on Robotics and Automation (ICRA)*, May 2008, pp. 3557–3563.

[17] J. Kato, D. Sakamoto, M. Inami, and T. Igarashi, "Multi-touch interface for controlling multiple mobile robots," in *Proc. of Int. Conf. Ext. Abstracts on Human Factors in Computing Systems*, 2009, pp. 3443–3448.

[18] A. Kolling, S. Nunnally, and M. Lewis, "Towards human control of robot swarms," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE Int. Conf. on*. IEEE, 2012, pp. 89–96.

[19] F. D. Crescenzio, G. Miranda, F. Persiani, and T. Bombardi, "A First Implementation of an Advanced 3D Interface to Control and Supervise UAV (Uninhabited Aerial Vehicles) Missions," *Presence: Teleoperators and Virtual Environments*, vol. 18, no. 3, pp. 171–184, June 2009.

[20] I. Maza, F. Caballero, R. Molina, N. Pea, and A. Ollero, "Multimodal Interface Technologies for UAV Ground Control Stations," *J. of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 371–391, 2010.

[21] J. Cooper and M. A. Goodrich, "Towards combining UAV and sensor operator roles in UAV-enabled visual search," in *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE Int. Conf. on*. ACM, 2008, pp. 351–358.

[22] D. Perez, I. Maza, F. Caballero, D. Scarlatti, E. Casado, and A. Ollero, "A Ground Control Station for a Multi-UAV Surveillance System," *J. of Intelligent and Robotic Systems*, vol. 69, no. 1-4, pp. 119–130–130, 2013.

[23] Y. Song, D. Demirdjian, and R. Davis, "Tracking body and hands for gesture recognition: Natops aircraft handling signals database," in *Proc. of Conf. on Automatic Face and Gesture Recognition*, March 2011, pp. 500–506.

[24] M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann, "A prototyping environment for interaction between a human and a robotic multiagent system," in *Proc. of the Int. Conf. on Human-Robot Interaction, (HRI)*, 2012, pp. 185–186.

[25] G. Jones, N. Berthouze, R. Bielski, and S. Julier, "Towards a situated, multimodal interface for multiple UAV control," in *Robotics and Automation (ICRA), 2010 IEEE Int. Conf. on*, 2010, pp. 1739–1744.

[26] T. Naseer, J. Sturm, and D. Cremers, "Followme: Person following and gesture recognition with a quadrocopter," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS'13)*, Tokyo, Japan, November 2013.

[27] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conf. on*, 2012, pp. 2815–2821.

[28] T. Krajník, V. Vonásek, D. Fiser, and J. Faigl, "Ar-drone as a platform for robotic research and education," in *Proc. of Int. Conf. Research and Education in Robotics - Eurobot*, June 2011, pp. 172–186.

[29] "ALVAR: Virtual and Augmented Reality Library," Accessed July 2013. [Online]. Available: http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html

[30] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM Int. Symp. on*, 2007, pp. 225–234.

[31] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.

[32] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[33] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, vol. 2749, pp. 363–370.

[34] C. Martin, F.-F. Steege, and H.-M. Gross, "Estimation of pointing poses for visual instructing mobile robots under real-world conditions," *Robotics and Autonomous Systems*, vol. 57, pp. 174–185, 2010.