Autonomous target following with monocular camera on UAS using Recursive-RANSAC tracker

Jae H. Lee[1], Jeff D. Millard[1], Parker C. Lusk[1] and Randal W. Beard[2]

***Abstract*— This paper presents a vision-based target tracking and following system using a monocular camera on an Unmanned Aerial System (UAS). The R-RANSAC tracker tracks multiple moving objects in the camera field of view and the proposed controller is capable of following a particular target selected by a user while keeping the target in the center of the image. The main contribution of this paper is that multiple objects can be tracked without imposing restrictions such as color, shape, etc. Also, the hardware test shows that the system is able to follow a target autonomously in a real-world outdoor environment. The proposed algorithm is validated on a 3DR X-8 multirotor platform using a downward facing camera.**

I. INTRODUCTION

Unmanned Aerial Systems (UAS) have become a popular platform for both research in academia and civil applications like filming, search and rescue, surveillance, and entertainment. UAS are advantageous for surveillance and target tracking because better visual awareness can be achieved with an airborne camera. Cameras are the most popular sensor on UAS because of cost, weight, and because they are a rich source information. For this reason, vision-based target tracking on UAS is an active area of research.

Vision-based target tracking has been studied for decades. For example, fixed-wing applications are found in [1], [2], [3], [4], [5], and multirotor applications are in [6], [7], [8], [9]. Tracking using a gimbaled camera is studied in [2], [3], [10], and tracking with a fixed camera is studied in [1], [5], [6], [7], [8], [9]. However, a common assumption that those studies make is that some information about the target, such as color [8], [9], shape [7] or pattern [11] is known or provided to the tracking algorithm. Thus, they require the user to specify what to track, or to provide the algorithm with a template image of the target in order for the tracker to be activated.

For example, the work in [1] demonstrates that a target can be kept in the camera field of view by constraining the roll angle of a small fixed-wing UAV in the presence of wind. However, the tracking method uses artificial color information and assumes that the target is static in the world frame. In [4], the tracking algorithm utilizes zero-mean normalized cross correlation to detect and locate the object of interest in the image, and therefore needs to be initialized by the user drawing a box around the target or with a template image of the target. Alternatively, the system described in [6] can follow any user specified target in an outdoor environment while the UAS maintains fixed distance to the target using the OpenTLD tracker [12]. Occlusions are also well handled due to the machine learning algorithm of the OpenTLD. The advantage of the tracker in [12] is that it does not require any previous knowledge about the target of interest and is able to track a great variety of objects. However, the system in [6] is strictly designed to track only one target at a time and needs a different tracking framework to extend to the multiple target tracking scenario. Also, the user has to draw a bounding box to initialize the track while trying not to include much of the background. Alternatively, reference [7] shows impressive results in following a fast moving target using a receding-horizon control scheme that minimizes the velocity error during the initial transience. Still, the system in [7] is limited to detecting and localizing spherical shaped objects of known size.

The work presented in this paper overcomes many of these limitations and assumptions by using the recursive random sample consensus (R-RANSAC) algorithm that was first introduced in [13] and that was developed to track multiple dynamic targets in clutter. The algorithm has been applied to problems like RADAR tracking [14], [13] and UAV Sense and Avoid [15]. It has also been applied to vision based scenarios in which R-RANSAC is used to track multiple moving objects from a camera mounted on both static and mobile platforms [16], [17]. This paper extends our previous work and is the first attempt to close the feedback loop of a UAS around the R-RANSAC vision based tracking algorithm.

The system presented here is unique in terms of tracking multiple objects that are in the camera field of view. Also, through the hardware demonstration, it is validated that the system can track realistic targets in unstructured environments with satisfactory performance. All operations in the hardware result are autonomous except for selecting the desired target ID. The control law presented in this paper is relatively simple and limited to a single target, but will be extended to multiple target tracking scenarios in future work. An overview of the system is found in Section II. Section III overviews the R-RANSAC tracking algorithm. The control strategy and hardware results are presented in Section IV and Section V. Finally, the conclusion is in Section VI.

*This research was supported by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation-sponsored industry/university cooperative research center (I/UCRC) under NSF Award No. IIP-1161036 along with significant contributions from C-UAS industry members, and in part by AFRL grant FA8651-13-1-0005.

[1]Jae Lee, Jeff Millard and Parker Lusk are graduate students in the Electrical and Computer Engineering department at Brigham Young University, Provo, UT 84602, USA jhl48@byu.edu, plusk@byu.edu

[2]Randal Beard is a professor of the Electrical and Computer Engineering department at Brigham Young University, Provo, UT 84602, USA beard@byu.edu

II. SYSTEM OVERVIEW

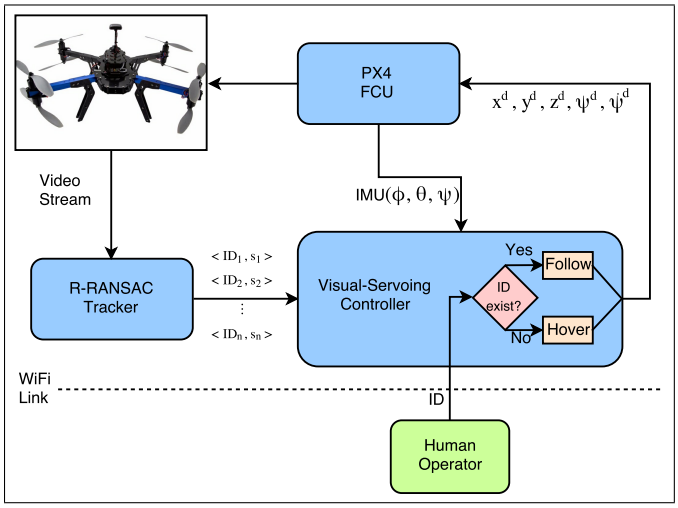


Fig. 1: System Architecture. The R-RANSAC tracker produces a set of target ID numbers and corresponding pixel locations. The visual-servoing controller outputs the desired position, heading, and yaw rate based on the pixel location of the requested target.

The R-RANSAC tracker and the visual-servoing controller are major subsystems as shown in Figure 1 and they communicate using the Robot Operating System (ROS) framework. The Pixhawk flight controller is used with the PX4 firmware for the autopilot. The R-RANSAC tracker is responsible for tracking moving objects in the image sequence and outputs a vector of normalized image coordinates with unique ID assigned to each track. Let the normalized pixel coordinates be defined as

$$s = [\epsilon_x, \epsilon_y] \tag{1}$$

where ϵ_x, ϵ_y are normalized pixel coordinates. Combined with IDs, a vector of track information is defined as

$$T = [< ID_1, s_1 >, < ID_2, s_2 >, \ldots, < ID_n, s_n >]. \tag{2}$$

The human operator assigns which target the UAV is to follow by sending a target ID number using the ground station. The controller checks to see if the target with the same ID given by the human operator exists among tracks. If the target exists, the controller keeps the target in the center of the image by commanding yaw rate and forward, backward motion of the UAV. Otherwise, the controller holds the UAV's current position until it receives another target ID existing among tracks.

III. R-RANSAC TRACKER

This section describes the visual detection and tracking framework. The objective of the visual tracker is to reliably track all targets in the field of view such that the ground operator can select a desired ID number for visual servoing. All elements of target tracking are required to be autonomous, without a priori knowledge of the number of targets in the field of view. A key requirement is track continuity (persistent track ID numbers) in order for the system to achieve good following performance. An ID-loss event requires the ground operator to select the new target ID when the track is re-initialized which leads to undesirable flight behavior. No detection aids such as color segmentation or truth data are available to the controller, meaning that target detection and state estimation must be robust in standard flight environments.

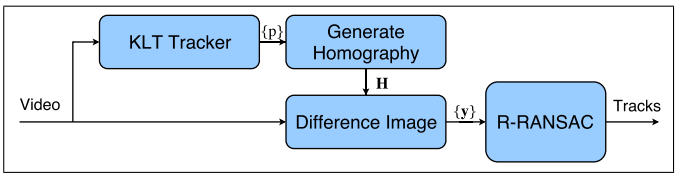


Fig. 2: This figure illustrates the detection framework used to generate measurements used by R-RANSAC. The KLT tracker creates point correspondences between frames which are used to calculate a homography. The difference image detects motion in the frame and creates position measurements.

Difference imaging reveals motion in the field of view by warping the previous image into the current timestep and comparing the two frames. This approach was used because it tends to be more robust in the presence of noisy homography transforms and image imperfections experienced by rolling shutter cameras in the presence of vibration. Our flight demonstration used entry-level hardware such as a webcam without gimbal stabalization.

This form of motion detection requires knowledge of the homography as seen in Figure 2. The KLT algorithm is used to create point correspondences accross the image and the homography is generated from these points using a RANSAC method.

R-RANSAC is an MTT algorithm that generates many hypothesis trajectories based upon an assumed dynamic model of the targets and the set of recent measurements. By elevating models that surpass a threshold of inlier measurements, the algorithm is capable of tracking many targets with missed detections in clutter [13].

For each new scan of measurements, the ones that are inliers to existing models are used to perform a Kalman update using a probabalistic data association (PDA) filter. For each measurement that is an outlier to all existing models, a new model is generated by sampling trajectories based on the recent history of recieved measurements. The sampled trajectory with the most support (having the most inliers) is selected to define the new model and the inlier measurements are used to estimate the target state estiamate and error covariance for the current timestep. Additional operations perform model merging and pruning in order to eliminate unlikely models.

By using the difference image measurements and R-RANSAC, track ID numbers are produced for targets in the field of view and can be used for visual servoing operations.

IV. UAV CONTROL

The visual tracking system in the previous section provides the control algorithm with a vector of normalized image coordinates for every track in the camera field of view. The control algorithm activates follow mode when there exists a target with the ID that a human operator has assigned for following. When the given ID is not found in the vector that the tracker provides, the control algorithm commands the UAV to hold its position until another target ID that exists among the tracks is assigned to follow. In this section, the control algorithm is described in more detail.

A. Coordinate Frame Convention

Before giving a detailed explanation of the control algorithm, it is worth clarifying our assumptions and the coordinate frames used.

First, an East-North-Up (ENU) coordinate frame is used as opposed to the common North-East-Down (NED) coordinates for UAV [18] in order to match the frame convention used in the `mavros` package in ROS [19]. Let $\mathcal{F}^i$ be the inertial frame, which in this case coincides with the ENU frame, and let $\mathcal{F}^v$ be the vehicle frame that is translated to the UAV center of mass, with the same orientation as $\mathcal{F}^i$. Vehicle-1 frame, $\mathcal{F}^{v1}$ indicates the frame that is only rotated about the z-axis of $\mathcal{F}^v$ by ψ, the heading angle of the multirotor. The rotation matrix from $\mathcal{F}^v$ to $\mathcal{F}^{v1}$ can be expressed as R_v^{v1}. Other involved frames are optical, camera, and body frames expressed as $\mathcal{F}^o$, $\mathcal{F}^c$, $\mathcal{F}^b$, respectively.

Second, a flat-earth model is used to properly scale the target position relative to the camera in $\mathcal{F}^{v1}$ and we have access to the correct altitude information.

Third, the displacement between the center of mass of the UAV and the focal point of camera is ignored since it is negligible compared to the distance between the camera and the target.

Fourth, we rely on GPS position controller on autopilot. The visual-servoing controller in this paper computes the desired multirotor position in order to follow the target and send the position command to the autopilot.

B. Forward and heading motion control

The first step of the motion control is to transform the line of sight (LOS) vector to the target in $\mathcal{F}^o$ into $\mathcal{F}^{v1}$. Let

$$\ell^o = [\epsilon_x^o, \epsilon_y^o, 1]^\top \tag{3}$$

where ℓ^o is the normalized line of sight vector in $\mathcal{F}^o$ and its third element 1 indicates the focal length of the camera in the normalized image. In our case, this is the information provided by the R-RANSAC tracker. Let also the unit vector along the optical axis in $\mathcal{F}^o$ be defined as

$$\mathbf{m}^o = [0, 0, 1]^\top. \tag{4}$$

By applying sequential transformations to ℓ^o and $\mathbf{m}^o$, we get

$$\ell^{v1} = R_b^{v1}(\phi, \theta) R_c^b(\alpha) R_o^c \ell^o = [\ell_x^{v1}, \ell_y^{v1}, \ell_z^{v1}]^\top \tag{5}$$

$$\mathbf{m}^{v1} = R_b^{v1}(\phi, \theta) R_c^b(\alpha) R_o^c \mathbf{m}^o = [m_x^{v1}, m_y^{v1}, m_z^{v1}]^\top \tag{6}$$

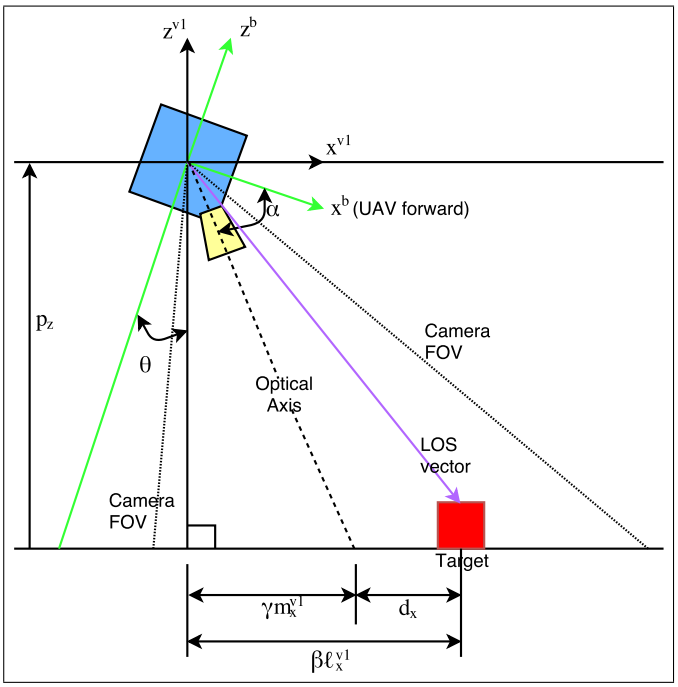


Fig. 3: Side view of the multirotor.

where R_c^b is a matrix with fixed values depending on how the camera is mounted with respect to $\mathcal{F}^b$, R_b^{v1} is a matrix requiring the roll and pitch angles of the multirotor. The ℓ^{v1} is the displacement of the target relative to the multirotor and $\mathbf{m}^{v1}$ is the optical axis in $\mathcal{F}^{v1}$. The LOS vector ℓ^{v1} and $\mathbf{m}^{v1}$ do not have proper scalings due to the unknown depth information to the target in $\mathcal{F}^o$, but can be recovered using the altitude of the camera. Let

$$\beta = \frac{p_z}{\ell_z^{v1}} \tag{7}$$

$$\gamma = \frac{p_z}{m_z^{v1}} \tag{8}$$

where p_z is the altitude of the multirotor. Then, the desired forward position from the current multirotor position can be computed as

$$d_x = \beta \ell_x^{v1} - \gamma m_x^{v1}. \tag{9}$$

This d_x may be further broken down into east and north components

$$d_n = d_x \sin(\psi) \tag{10}$$

$$d_e = d_x \cos(\psi) \tag{11}$$

where ψ is the heading of the multirotor. These north and east components are added to the current multirotor east and north positions, and the sum is sent to the autopilot position controller.

It is more suitable to compensate for a target moving horizontally in the image plane by adjusting the multirotor's heading than through lateral motion. Thus, a yaw rate command ω_z can be computed as

$$\omega_z = \eta \ell_y^{v1}, \tag{12}$$

where $\eta > 0$ is a control gain.

V. Experiments and Results

The main hardware is comprised of a 3DR X8 multirotor platform, a small form factor Gigabyte BRIX GB-BXi7-4500 (no GPU), low-cost USB camera, ELP-USBFHD01M-L21 (rolling shutter), and Pixhawk with PX4 firmware. The proposed system was tested in an outdoor environment to track realistic targets (people). We demonstrate the ability to follow one of the multiple tracked objects while switching the target of interest in real-time. The tracker does not have any prior information about what tracks look like and how they might move. The hardware result shows that the R-RANSAC tracker is able to track multiple targets and to provide the controller with proper coordinates of the targets. It also shows that the controller is able to follow one of the target while keeping it in the camera field of view.

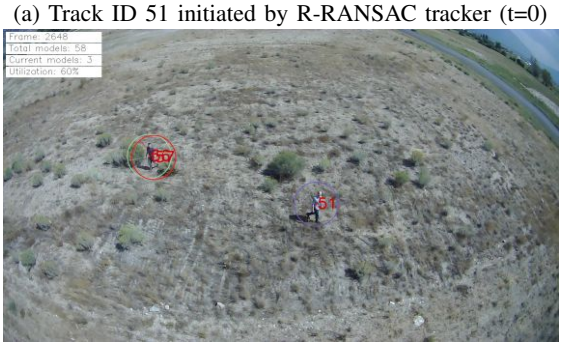
As shown in Figure 4a, the human operator sees this camera view of multiple moving objects being tracked from the ground station and commands the multirotor to follow a target of interest by sending the correct track ID (Figure 4b). The hardware test lasted about 1 minute and in the middle (t=35) the operator commanded the multirotor to follow a different target (Figure 4d). Until another track ID given from the operator, the multirotor keeps following the current assigned target (Figure 4e).

Figure 5 and 6 show the effort of the multirotor trying to place the target being followed in the center of image plane. In Figure 5 and Figure 6, the numbered events listed in Figure 4 are illustrated in the image plane. A track with ID 51 is initialized by the R-RANSAC tracker, and later the multirotor was commanded to follow track 51. The R-RANSAC tracker detected another moving object in the camera field of view and started to track the object with ID 65 while the multirotor was still following the track 51. After a while, the human operator switches the target of interest resulting in the multirotor following track 65. The multirotor kept following the track 65 for the rest of experiment. Finally, Figure 7 illustrates the GPS coordinates of the multirotor to show its movement and heading during the experiment.

VI. Conclusion

In this work, a novel vision-based target following system with the R-RANSAC tracker is presented with hardware demonstration. The experimental result shows the feasibility of the real-time system in a realistic outdoor environment. With the R-RANSAC tracker, multiple moving objects in the camera view are tracked without having to know their colors or shapes. The controller is able to follow any particular target among the tracks with minimum effort to the human operator. The human operator is only expected to send a track ID number to the controller in order for the multirotor to follow the target of interest. This research opens up many other potential areas of research such as keeping multiple targets in the camera field of view, human machine interaction and multi UAS coordination in multiple target tracking situations.

(a) Track ID 51 initiated by R-RANSAC tracker (t=0)



(b) Human operator commanded to follow track ID 51 (t=13)

(c) Track ID 65 initiated by R-RANSAC tracker (t=27)

(d) Human operator commanded to follow track ID 65 (t=35)

(e) A snapshot of the track ID 65 being followed (t=60)

Fig. 4: Camera view at various events during t=0-60.

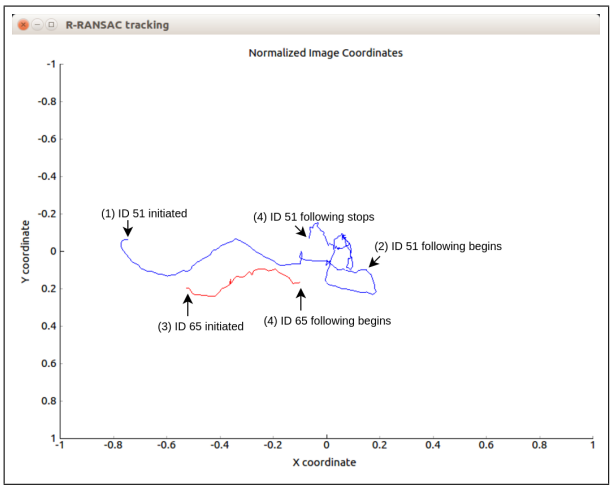


Fig. 5: Tracks movement in the normalized image plane. Each event (1)-(4) corresponds to camera view in 4a-4d respectively. Until the command to follow ID 65, the multirotor keeps the track ID 51 from leaving the camera view.

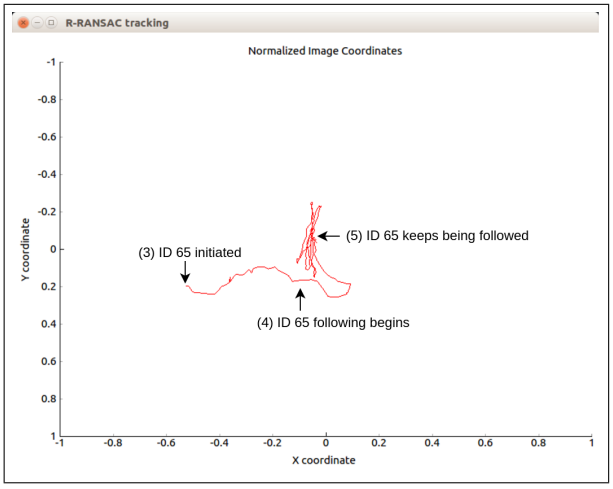


Fig. 6: The movement of track ID 65 in the normalized image plane. Each event (3)-(5) corresponds to camera view in 4c-4e respectively. The controller keeps the track ID 65 in the camera field of view after receiving the command to do so from the human operator.

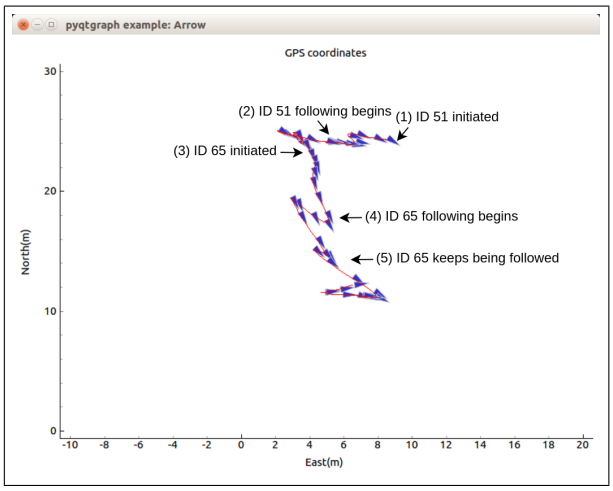


Fig. 7: Multirotor GPS footage and heading corresponding to camera view in 4a-4e respectively.

References

[1] J. Saunders and R. W. Beard, "Visual Tracking in Wind with Field of View Constraints," *International Journal of Micro Air Vehicles*, vol. 3, pp. 169–182, sep 2011.

[2] R. Rysdyk, "Unmanned Aerial Vehicle Path Following for Target Observation in Wind," *Journal of Guidance, Control, and Dynamics*, vol. 29, pp. 1092–1100, sep 2006.

[3] V. Dobrokhodov, I. Kaminer, K. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small UAVs," in *2006 American Control Conference*, p. 6 pp., IEEE, 2006.

[4] A. Qadir, J. Neubert, W. Semke, and R. Schultz, "On-Board Visual Tracking With Unmanned Aircraft System (UAS)," in *Infotech@Aerospace 2011*, (Reston, Virigina), p. 9, American Institute of Aeronautics and Astronautics, mar 2011.

[5] P. Theodorakopoulos and S. Lacroix, "A strategy for tracking a ground target with a UAV," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1254–1259, IEEE, sep 2008.

[6] J. Pestana, J. L. Sanchez-Lopez, P. Campoy, and S. Saripalli, "Vision based GPS-denied Object Tracking and following for unmanned aerial vehicles," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–6, IEEE, oct 2013.

[7] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar, "Autonomous Flight for Detection, Localization, and Tracking of Moving Targets With a Small Quadrotor," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1762–1769, jul 2017.

[8] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying UAV," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4929–4934, 2011.

[9] J. Kim and D. H. Shim, "A vision-based target tracking control system of a quadrotor by using a tablet computer," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1165–1172, IEEE, may 2013.

[10] Z. Hurak and M. Rezac, "Image-Based Pointing and Tracking for Inertially Stabilized Airborne Camera Platform," *IEEE Transactions on Control Systems Technology*, vol. 20, pp. 1146–1159, sep 2012.

[11] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *2012 IEEE International Conference on Robotics and Automation*, pp. 971–976, IEEE, may 2012.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1409–1422, jul 2012.

[13] P. C. Niedfeldt and R. W. Beard, "Multiple target tracking using recursive RANSAC," in *2014 American Control Conference*, pp. 3393–3398, IEEE, jun 2014.

[14] E. B. Quist, P. C. Niedfeldt, and R. W. Beard, "Radar odometry with recursive-RANSAC," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1618–1630, 2016.

[15] J. Wikle and T. W. Mclain, "Detection and tracking of multiple intruders using the recursive-ransac algorithm state of practice." Utah Space Grant Fellowship Symposium, 2012.

[16] K. Ingersoll, "Vision based multiple target tracking using recursive ransac," Master's thesis, Brigham Young University, Provo, UT, 2015.

[17] P. Defranco, "Detecting and tracking moving objects from a small unmanned air vehicle," Master's thesis, Brigham Young University, Provo, UT, 2015.

[18] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft Theory and Practice*. Princeton, NJ: Princeton, 2012.

[19] V. Ermakov, "mavros." http://wiki.ros.org/mavros.