Vision-Based Target Tracking and Autonomous Landing of a Quadrotor on a Ground Vehicle

Tru Hoang[1], Enkhmurun Bayasgalan[1], Ziyin Wang[2], Gavriil Tsechpenakis[2], Dimitra Panagou[1]

***Abstract*— This paper addresses vision-based tracking and landing of a micro-aerial vehicle (MAV) on a ground vehicle (GV). The camera onboard the MAV is mounted so that the optical axis is aligned with the downward-facing axis of the body-fixed frame. A novel supervised learning vision algorithm is proposed as the method to detect the ground vehicle in the image frame. A feedback linearization technique is developed for the MAV to fly over and track the GV so that visibility with the tracked target is maintained with certain guarantees. The efficacy of the visual detection algorithm, and of the tracking and landing controller is demonstrated in simulations and experiments with static and mobile GV.**

I. Introduction

Recent advances in computer vision and the decreasing cost of visual sensors have enabled aerial surveillance and mapping [1]–[3] navigation in GPS-denied environments [4]–[6], and state estimation using these sensors [8]. Another scenario of interest is the tracking of one or multiple ground targets using a team of MAVs. Here we focus on the design of a vision-based controller that achieves tracking a GV and autonomous landing of a quadrotor on the tracked target.

Recently, the problem of vision-based tracking has been addressed in a growing body of literature; see [9]–[13], [16], [17] and references therein. In [9], Teulière *et al.* proposed to apply proportional control for both the translational velocities and the yaw rate of the MAV. Similarly, Nakamura *et al.* used PID controller for the target tracking task [10]. Gomez-Balderas *et al.* [11] presented a hybrid controller that addresses MAV guidance for three different operating modes: take-off, target tracking, and loss of moving target. Each mode has two control laws: an integral sliding mode controller to perform attitude control and a PI controller for position control. In [12], Bohdanov proposed an LQG controller for waypoint tracking, and extended this controller to a mobile target by modeling the moving waypoint as a Gaussian process. Lee *et al.* [13] addressed the autonomous landing by using image-based visual servoing to generate translational and rotational velocity references and an adaptive sliding mode controller to track these inputs. Hérissé *et al.* [14] proposed a PI-type nonlinear tracking and landing controller using optical-flow approach. Serra *et al.* [15] proposed a P controller augmented with a disturbance estimator for landing on a textured platform. A common theme amongst these works is the explicit assumption that the target velocity is constant.

In [16], a nonlinear tracking controller was developed for a fixed-wing MAV with a gimbaled camera as its visual sensor. Zhang and Liu [17] developed a tracking law using the Lyapunov vector field approach for a similar system configuration. Although the ability to control the camera FOV independently of the MAVs motion decreases the probability of a tracking-loss event, not all available commercial MAVs can support the required payload of the gimbal. Moreover, its inclusion can decrease the operating time for the tracking task by incurring energy overhead on the overall system.

In this paper we propose a novel object-detecting vision algorithm for vision-based target tracking, and a feedback linearization controller for the autonomous target-following and landing for a MAV. Compared to the aforementioned papers, here we provide a control approach that is based on feedback linearization, and does not assume a detailed target motion model. Now, in visual object recognition there are several methods that achieve satisfactory results [22]–[24], [26], [27], yet usually they cannot be used for online (real-time) detection and recognition, due to computational complexity. In our approach, we calculate SIFT [30] features, and employ the Bag-of-Visual-Words [22] representation, instead of applying segmentation or hierarchical construction of object proposals [28], [29]. As we show here, our object detection method produces the desired results efficiently and robustly during the quadrotor flight.

The remainder of the paper is organized as follows. Section II briefly introduces the vision algorithm used to detect a known target in real time. The tracking model derived from the information obtained from the vision algorithm is shown in detail in Section III. The proposed control law is presented in Section IV. The efficacy of the control law is demonstrated through simulation results in Section V. The experimental setup used to evaluate the performance of the control law on a real system is presented in Section VI-.1 and the results are subsequently shown in Section VI-.2. Finally, this paper concludes with some remarks in Section VII.

II. Visual Target Detection

We consider a number of objects of interest, from predetermined categories, that we want to detect during the quadrotor flight.

[1]Aerospace Engineering Dept., University of Michigan, Ann Arbor, T. Hoang: truhoang@umich.edu, E. Bayasgalan: murunb@umich.edu, D. Panagou: dpanagou@umich.edu

[2]Computer and Information Science Dept., Indiana U.-Purdue U, Indianapolis, Z. Wang: ziyiwang@umail.iu.edu, G. Tsechpenakis: gavriil@cs.iupui.edu

Dimitra Panagou would like to acknowledge the support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-14-2-0001 U.S. Army TARDEC in Warren, MI.

Tru Hoang would like to acknowledge the support by the Rackham Graduate School at the University of Michigan, Ann Arbor through the Rackham Summer Award.

This is the author's manuscript of the article published in final edited form as:
Hoang, T., Bayasgalan, E., Wang, Z., Tsechpenakis, G., & Panagou, D. (2017). Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle. In 2017 American Control Conference (ACC) (pp. 5580–5585). https://doi.org/10.23919/ACC.2017.7963823

Off-line training. For each object category, we acquire video samples with a moving camera, depicting the object in a smooth (uncluttered) background. Such videos illustrate different views of the target, with respect to off-plane rotation and scale. Each frame of each video is a training instance (sample), and each video corresponds to a class, namely an object category. During training, we use the Bag-of-Visual-Words [22] framework, which consists of:

- Detect SIFT [30] features from each image
- Cluster the detected SIFT descriptors into k centers using the K-means++ [31] algorithm; these k centers are the 'visual vocabulary'.
- Encode each image as a histogram of k visual vocabulary entities; this way, each image is described by an $1 \times k$ vector.
- Train a normal-invert-Wishart classifier [32] on those encoded images ($1 \times k$ vectors).

In practice, to improve the trade-off between training speed and accuracy, we do not consider non-informative (background) features by tuning edge and point thresholds in the SIFT detection algorithm. Also, the normal-invert-Wishart classifier is a probabilistic generative model that is able to return a reasonable likelihood for each tested sample instance, and allows for adding more object categories without retraining.

On-line target detection. During a quadrotor flight, the camera captures multiple objects within the view field. The task of the on-line detection is to find candidate objects ('object proposals') in the streaming images that we can encode and classify. For each input frame the system performs the following steps:

- Detect SIFT features and make assignments according to the visual vocabulary calculated off-line.
- Produce object proposals based on the spatial location of the detected features, as explained below.
- For each object proposal, compute the likelihoods of belonging to the pre-determined categories.
- If the highest likelihood for the 'proposed' target is higher than a threshold, we send message to the control system along with the target location on image plane.

1) Object Proposals Based on SIFT Features: Since we aim at detecting objects in real time (during frame acquisition), popular segmentation-based object proposal methods, such as [28], [29], are not suitable. To avoid segmentation and maintain low complexity, we follow a fast object proposal approach that uses the spatial locations of SIFT features and parses each location once. The basic idea is that the detected local features, such as corners and edges, are spatially dense in informative regions, i.e., where there is an object of potential interest. Under this assumption, spatially sparse features are considered as background or noise.

Fig. 1 describes our approach. Algorithmically, each feature f is the x-y coordinates of a detected SIFT descriptor in the image plane. The algorithm maintains two sets when parsing the features. Set O corresponds to descriptor locations inside detected object regions, which will be returned as the final result. Set T stores temporary location features, which could be confirmed and moved to O if they are matched frequently, or discarded as noise. For every entity in both O and T we also store a counter that indicates how many times the entity has been matched with input features.

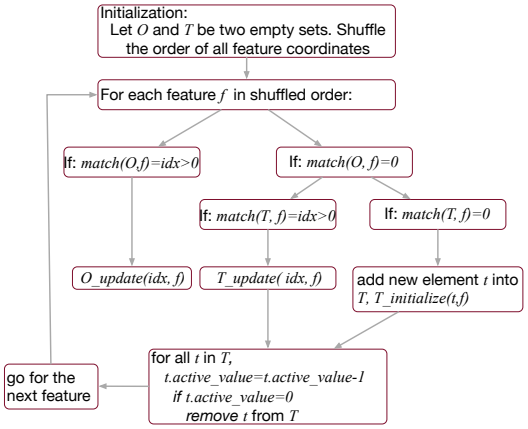

Fig. 1. Algorithmic illustration of our object proposal approach (see text).

For every instance of the algorithm (for each input f), $match(O, f)$ computes the Euclidean distance between f and all elements in set O. If the smallest distance is smaller than or equal to a user specified threshold θ, $match(O, f)$ returns the index (position in O) of that element, idx. If the smallest distance is larger than θ, $match(O, f)$ returns 0.
(a) If we find a match for f in the set O, we assign f to the position idx in O, calculate/update the center of the population in position idx, and update the corresponding matching frequency counter with routine $O\_update(idx, f)$.
(b) If we do not find a match for f in O, $match(O, f) = 0$, we use the routine $match(T, f)$ for matching in the temporary set T. If a match is found, we update T with $T\_update(idx, f)$ (similar to $O\_update(idx, f)$). If no match is found, $T\_initialize(t, f)$ creates a new element t in T where f is stored.

As we explain above, frequently matched entities in T are moved to the set O as informative. We use the notion of 'active value', $active\_value$, for every entity t in T to decide whether it should move to O as a new entity (emerged new informative location in the image), or it should be permanently discarded as noise. Initially, every new entity in T has $active\_value = a_0 > 0$. Every time an entity t is matched, its active value $t.active\_value$ increases by 1, otherwise it reduces by 1. If $t.active\_value > \phi > a_0$, the entity moves to O, with ϕ being a positive constant used as decision threshold. Before a new feature is processed, $active\_value$ is reduced by 1 for all elements in T. When $t.active\_value = 0$, t is permanently discarded.

III. Mathematical Model

Using the target detection framework described in the previous section, we develop a mathematical model that allows the use of information provided by the tracking

algorithm for feedback control design. It must be noted that this information is limited to the pixel coordinates of the tracked object in the image plane. We address this limitation by assuming that the target travels on a flat terrain, thereby allowing us to use the altitude of MAV as a depth estimate.

The derivation of the model assumes that target is initially in the FOV and the image plane is always parallel to the ground, i.e., pitch angle (θ) and roll angle (ϕ) are equal to zero, allowing the choice of $\mathbf{u}$ as the linear velocities of the quadrotor resolved in $\mathcal{B}$, and $\dot{\psi}$ as its yaw rate.

Let $\mathbf{x}^{\mathcal{W}}$ and $\mathbf{x}_T^{\mathcal{W}}$ denote the position vectors of the quadrotor and the target resolved in $\mathcal{W}$ (world frame), respectively. Then, the position error vector is defined as $\mathbf{e}^{\mathcal{W}} \coloneqq \mathbf{x}_T^{\mathcal{W}} - \mathbf{x}^{\mathcal{W}}$. This implies that:

$$\dot{\mathbf{e}}^{\mathcal{W}} = \dot{\mathbf{x}}_T^{\mathcal{W}} - \dot{\mathbf{x}}^{\mathcal{W}}. \tag{1}$$

Let $\mathbf{e}^{\mathcal{I}}$ denote the error vector in the image frame $\mathcal{I}$, expressed in homogeneous coordinates, and ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}$ the rotation matrix between frames $\mathcal{W}$ and $\mathcal{B}$. Following the pinhole camera model in [18], we have:

$$\lambda \mathbf{e}^{\mathcal{I}} = \mathbf{K}\left({}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\mathbf{e}^{\mathcal{W}}\right), \tag{2}$$

where $\mathbf{K}$ is the *intrinsic parameter matrix*, which contains the focal length, scale factor, and skew factor of the camera and λ is the depth of the object from the optical center. In our case, λ is taken to be the altitude of the MAV.

Differentiating (2) with respect to (w.r.t.) time, we get:

$$\lambda \dot{\mathbf{e}}^{\mathcal{I}} = \mathbf{K}\left({}^{\mathcal{B}}\dot{\mathbf{R}}_{\mathcal{W}}\mathbf{e}^{\mathcal{W}} + {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\dot{\mathbf{e}}^{\mathcal{W}}\right). \tag{3}$$

Using the expression of $\dot{\mathbf{e}}^{\mathcal{W}}$ in (1) and the result of differentiating a rotation matrix presented in [19], which relates time differentiation of the rotation matrix to the angular velocities of a rigid body relative to $\mathcal{W}$, (3) becomes:

$$\lambda \dot{\mathbf{e}}^{\mathcal{I}} = \mathbf{K}\left[-\boldsymbol{\omega}^{\mathcal{B}\times}\left({}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\mathbf{e}^{\mathcal{W}}\right) + {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}(\dot{\mathbf{x}}_T^{\mathcal{W}} - \dot{\mathbf{x}}^{\mathcal{W}})\right], \tag{4}$$

where $\omega^{\mathcal{B}\times}$ is the skew-symmetric matrix of the angular velocity vector of the quadrotor $\omega^{\mathcal{B}}$ resolved in $\mathcal{B}$. Substituting (2) into (4) for $\mathbf{e}^{\mathcal{W}}$, we obtain:

$$\begin{aligned}\lambda \dot{\mathbf{e}}^{\mathcal{I}} = \mathbf{K}\{-\lambda\boldsymbol{\omega}^{\mathcal{B}\times}\left[{}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\left(\mathbf{K}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\right)^{-1}\mathbf{e}^{\mathcal{I}}\right] \\ + {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}(\dot{\mathbf{x}}_T^{\mathcal{W}} - \dot{\mathbf{x}}^{\mathcal{W}})\}.\end{aligned} \tag{5}$$

Define $\boldsymbol{\alpha} \coloneqq \mathbf{e}^{\mathcal{B}} = \lambda\mathbf{K}^{-1}\mathbf{e}^{\mathcal{I}}$ and $\mathbf{u} \coloneqq {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\dot{\mathbf{x}}^{\mathcal{W}}$. Then, the kinematic model in (5) is rewritten as:

$$\dot{\boldsymbol{\alpha}} = -\boldsymbol{\omega}^{\mathcal{B}\times}\boldsymbol{\alpha} + {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\dot{\mathbf{x}}_T^{\mathcal{W}} - \mathbf{u}. \tag{6}$$

Define $\varepsilon \coloneqq \eta_T - \psi$ as the heading error, where η_T is the heading angle of the target and ψ the heading angle of the MAV. It is then straightforward to obtain the following angular error kinematic model, where $\dot{\psi}$ is a control input:

$$\dot{\varepsilon} = \dot{\eta_T} - \dot{\psi}. \tag{7}$$

The complete kinematic model is given below:

$$\begin{bmatrix}\dot{\boldsymbol{\alpha}} \\ \dot{\varepsilon}\end{bmatrix} = \begin{bmatrix}-\boldsymbol{\omega}^{\mathcal{B}\times}\boldsymbol{\alpha} + {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\dot{\mathbf{x}}_T^{\mathcal{W}} \\ \dot{\eta}_T\end{bmatrix} - \begin{bmatrix}\mathbf{u} \\ \dot{\psi}\end{bmatrix}. \tag{8}$$

IV. Control Law

We propose a feedback linearization-based control law to guarantee that the error vector will be uniformly and ultimately bounded. The control law is given as:

$$\begin{bmatrix}\mathbf{u} \\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}\mathbf{k_v}\boldsymbol{\alpha} - \boldsymbol{\omega}^{\mathcal{B}\times}\boldsymbol{\alpha} \\ k_\psi\varepsilon\end{bmatrix}, \tag{9}$$

where $k_\psi > 0$, $\mathbf{k_v} = \begin{bmatrix}k_{\mathbf{v}}^x & 0 & 0 \\ 0 & k_{\mathbf{v}}^y & 0 \\ 0 & 0 & sk_{\mathbf{v}}^z\end{bmatrix}$, $s = \begin{cases}1, & t \geq \tau \\ 0, & t < \tau\end{cases}$.

The variable t stores the amount of time the quadrotor has successfully tracked the target. τ is a parameter greater than or equal to zero and its choice depends on the application, e.g. for surveillance of the ground vehicle for finite time before descent, a possible value for τ is 30 seconds. The introduction of s in the controller provides a way to schedule the activation of the descent of the quadrotor.

Since the model includes one unknown term, i.e., the target motion, the most that we can conclude is the boundedness of the error, provided a few assumptions be made regarding the unknown term [20]. We provide guarantees that the target remains in the camera FOV in the following proposition:

Proposition 1.
Let z denotes the error vector $\boldsymbol{\alpha}$ projected onto the image plane, i.e. $z \coloneqq L\boldsymbol{\alpha}$, where $L : \mathbb{R}^3 \to \mathbb{R}^2$ is a linear projection operator, and $d \coloneqq L^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\dot{\mathbf{x}}_T^{\mathcal{W}}$, the projected target velocity vector onto the same plane. Let $S \coloneqq \{z \in \mathbb{R}^2 \,|\, \|z\|_2 < r\}$, where $r = \frac{1}{2}min\{l, w\}$, l and w are the camera frame length and width in metric units, respectively. Suppose δ is an upper-bound on the target speed and $z(0) \in S$. If $\delta < cr$, for some constant c then the target will not escape the camera FOV, i.e., $z(t) \in S, \forall\ t \geq 0$.

Proof: Applying (9) to the system (8), the closed-loop system in the image plane can be expressed as:

$$\dot{z} = Az + d, \tag{10}$$

where $A = diag\{-k_{\mathbf{v}}^x, -k_{\mathbf{v}}^y\}$. Let V be a quadratic Lyapunov function represented by

$$V(z) = z^T P z, \tag{11}$$

where P is positive-definite matrix obtained from solving the Lyapunov equation $PA^T + AP + \gamma I = 0$, $\forall\gamma > 0$. V satisfies the following:

$$\begin{aligned}\lambda_{min}(P)\|z\|^2 \leq V \leq \lambda_{max}(P)\|z\|^2, \\ \frac{\partial V}{\partial z}Az = -\gamma\|z\|^2 \\ \|\frac{\partial V}{\partial z}\| = \|2z^T P\| \leq 2\|P\|\|z\| = 2\lambda_{max}(P)\|z\|.\end{aligned} \tag{12}$$

Since A is a diagonal of a vector of negative constants, P has the form $P = diag\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}$. Then the maximum and minimum eigenvalues of P are $\max\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}$ and $\min\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}$, respectively. From [20], with appropriate choice of $k_{\mathbf{v}}^x$ and $k_{\mathbf{v}}^y$ we can set c to

$$c = \frac{\gamma\theta}{2\max\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}}\sqrt{\frac{\min\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}}{\max\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}}}. \tag{13}$$

It follows that

$$\|z(t)\| \leq \frac{2\max\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}\delta}{\gamma\theta}\sqrt{\frac{\max\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}}{\min\{\gamma/2k_{\mathbf{v}}^x, \gamma/2k_{\mathbf{v}}^y\}}}$$
$$< r$$

■

In practice, one first determines δ for ones particular system and choose appropriate gains such that $\delta < cr$, where c is as shown in Eq. 13. Hence, for the remainder of the paper, we assume that the conditions in Proposition 1 hold.

V. Simulation

The behavior of the closed-loop system is simulated in MATLAB using QRSim [21]. QRSim is a simulation software package that provides realistic quadrotor dynamics and its onboard sensors (i.e. GPS, IMU, camera) with their stochastic inaccuracies. Two different scenarios where simulated; a static target, and a mobile target following a linear trajectory as the GVs. In both scenarios, it is assumed that there is no accelerometer, gyroscope, and barometric altimeter noise. Moreover, it is assumed that the GVs are always located in the camera FOVs at $t = 0$, and the descent controller is activated immediately, i.e., $\tau = 0$. We omit the results of the static target in the interest of space.

Fig. 2 shows the trajectory of the quadrotor (blue) chasing a mobile target (red). The target is initially located at the origin and starts moving with bounded velocity along a linear trajectory as $\|\dot{\mathbf{x}}_T^{\mathcal{W}}\|_2 \in \left[-0.5, 0.5\right]\ m/s$.

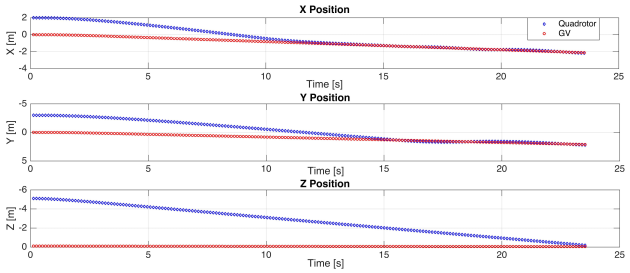

Fig. 2. Trajectories of the quadrotor and the mobile GV.

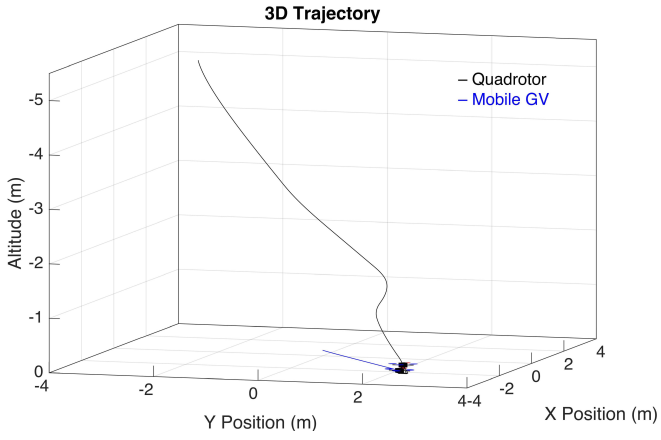

Fig. 3. 3D View of the trajectories of the quadrotor and the mobile GV.

$\mathbf{x}_0^{\mathcal{W}}$ is set as $(2, -3, -5)$. The vertical gain $k_{\mathbf{v}}^z$ is set as 0.007. Similar to the static target, it is assumed that $\tau = 0$, and the mobile target is to be located in the camera FOV at all times. Gains $k_{\mathbf{v}}^x$ and $k_{\mathbf{v}}^y$ were not changed and were set same as the previous part. The quadrotor velocity is bounded as $\|\dot{x}^{\mathcal{W}}\|_2 \in \left[-1.5, 1.5\right]\ m/s$. Finally, Fig. 3 shows the 3D paths of the vehicles.

VI. Experiments

1) Experimental Setup: The control law is validated experimentally in an indoor test environment. We used a commercially available Hummingbird quadrotor manufactured by Ascending Technologies. It features two embedded processors running at 1 kHz, denoted as the high-level processor (HLP) and the low-level processor (LLP), an IMU running at 100 Hz, a GPS receiver, and a barometer. The quadrotor is also equipped with a 1.91 GHz micro-computer running Linux and a USB camera with a maximum frame rate of 93 Hz and 58° FOV. The software is implemented via ROS . The camera was calibrated *a priori* to obtain $\mathbf{K}$ and samples images at 80 Hz. The vision algorithm processes the images at 80 Hz then sends the position of the GV to the controller which runs at 60 Hz. The velocity commands are computed and sent to the HLP via serial communication. The commands are converted into motor commands and relayed to the LLP for execution. The Vicon motion capture system was used to provide measurements on the MAV's yaw orientation and altitude in the $\mathcal{W}$ frame; estimating attitude via computer vision techniques is outside of the scope of this paper, while the altitude measurement from the barometer on board the MAV is very noisy. Estimating the full state vector of the MAV using information from the available onboard sensors only is ongoing work.

2) Results: We performed 6 trials for the static target tracking experiment, and 4 trials for the mobile target tracking with the quadrotor flying at an altitude of 1.5 meters. The results are shown in Fig. 4 and Fig. 6, respectively. The 3D trajectories of the MAV and GV in the last trial of the respective experiments are also plotted in Fig. 5 and Fig. 7.

In Fig. 4, all trials show satisfactory tracking performance although Trial 1 tracking controller was activated much later than the others. This is due to the displacement of the MAV during its climbing phase causing the GV to lie outside the onboard camera FOV when the MAV has reached its operating altitude. Once the GV was detected, the tracking controller was able to maintain target detection and drive the horizontal position error to zero after about 5 seconds. The descent controller was activated after t_{track} reached 30 seconds.

The 3D trajectory of the MAV and GV in Fig. 5 shows an offset between the MAV and the GV centers during tracking and a wider offset after the MAV has landed. The reason for the first offset can be explained by the hardware setup. The camera is mounted 0.095 meter away from the MAV center-of-gravity. This offset was only accounted for in the heading control since the actual heading error is smaller than perceived by the camera. The additional offset observed

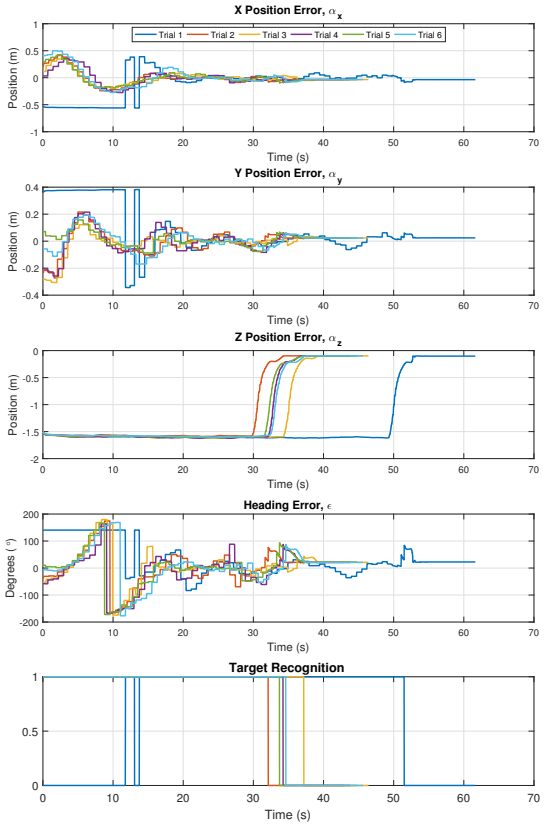

Fig. 4. Position errors and target recognition, represented by a binary variable, from 6 trials with a static target. The parameters used are $k_{\mathbf{v}}^x = 0.4$, $k_{\mathbf{v}}^y = 0.5$, $k_{\mathbf{v}}^z = 0.8$, $k_\psi = 0.3$, and $\tau = 30$ seconds.

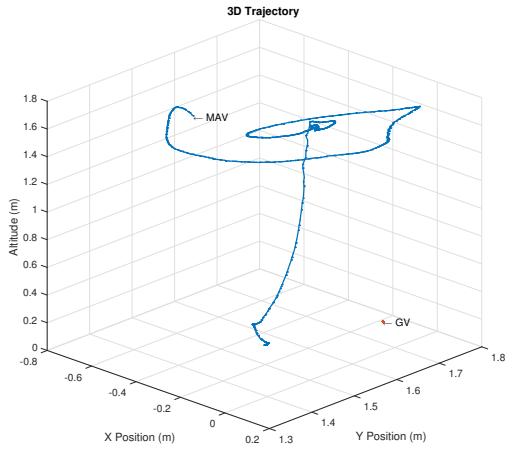

Fig. 5. Trial 6: The resulting 3D paths of the MAV and the GV.

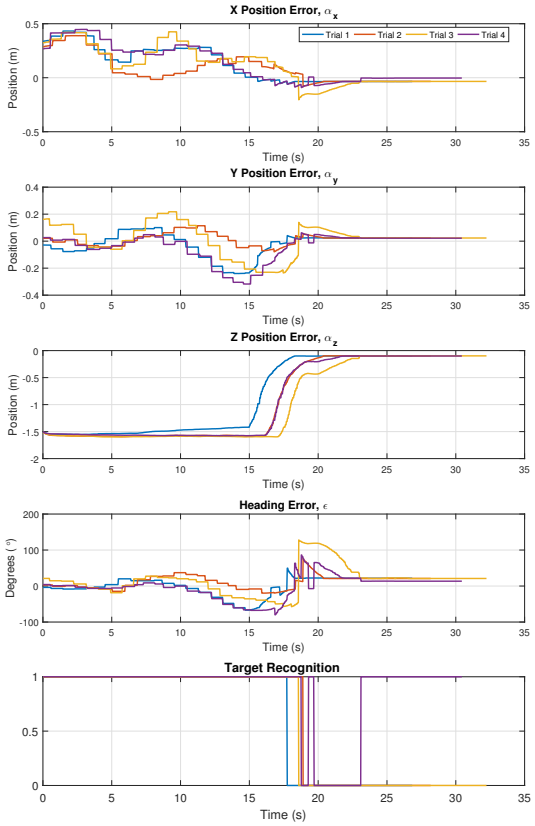

Fig. 6. Position errors and target recognition, represented by a binary variable, from 4 trials with a mobile target. The parameters used are$k_{\mathbf{v}}^x = 0.4$, $k_{\mathbf{v}}^y = 0.5$, $k_{\mathbf{v}}^z = 0.8$, $k_\psi = 0.3$, and $\tau = 15$ seconds.

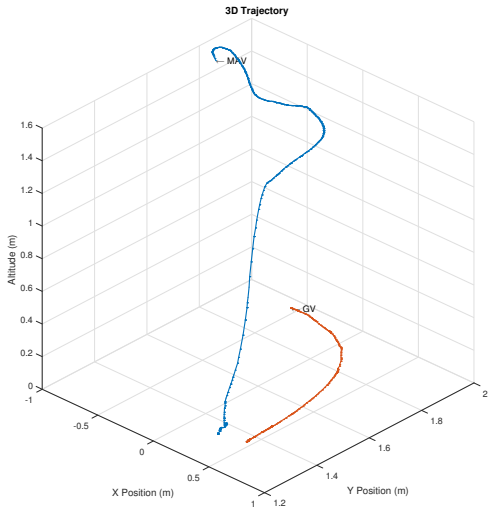

Fig. 7. Trial 4: The resulting 3D paths of the MAV and the GV.

during landing of the MAV can be explained by the ground effect. The tracking controller could not compensate for this additional error fast enough before the MAV reached the ground, due to the relatively large z gain.

The convergence of the position error for all of the trials shown in Fig. 6 demonstrates the tracking capability of the proposed control law for a slow moving mobile target. Trial 4 target recognition data shows that the vision algorithm was still able to detect the GV after the MAV has landed because the camera was directly on top of the GV and the vision algorithm was still able to detect the features.

Fig. 7 reconstruct the 3D trajectory of the MAV and the GV in the $\mathcal{W}$ frame in Trial 4. It shows that the MAV was able to maintain its position above the GV during the tracking phase and descended after 15 seconds.

The target velocities in $\mathcal{W}$ during the mobile target tracking experiment were captured and shown in Fig. 8. The maximum target speed for each trial is 2.056 m/s, 2.138 m/s,

2.131 m/s, and 1.980 m/s, respectively.

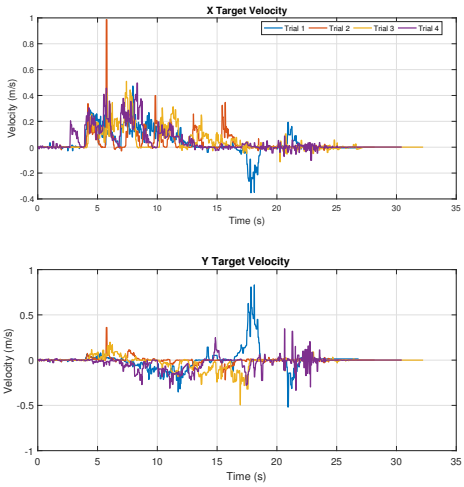

Fig. 8. Target velocity for the mobile tracking experiment. The velocity is computed by taking the derivative of the target's position in each update. z velocity is not shown because the terrain is flat.

VII. Conclusion

In this paper, we proposed a novel object-detecting vision algorithm and a vision-based tracking and autonomous landing controller using a feedback linearization technique, and demonstrated their effectiveness via experiments. We established the ultimate boundedness of the error vector by modeling the target motion as a nonvanishing perturbation. This implies that a slow-moving target does not escape the camera frame once the controller is engaged. The results verify this conclusion and show that the vision algorithm can provide the target's position consistently.

Future work can focus on developing an observer to estimate the target motion and feed this information into the feedback linearized controller for complete control of the system behavior. Another area of focus is the incorporation of a depth estimator into the vision algorithm for complete information of the target's position.

References

[1] Forster, C., Pizzoli, M., Scaramuzza, D., 'Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles,' in *Proc. of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[2] Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., Scaramuzza, D., 'Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle,' *Journal of Field Robotics*, vol. 33, pp. 4:1-20, 2015.

[3] Doitsidis, L., Weiss, S., Renzaglia, A., Achtelik, M., Kosmatopoulos, E., Siegwart, R., Scaramuzza, D., 'Optimal Surveillance Coverage for Teams of Micro Aerial Vehicles in GPS-denied Environments using Onboard Vision,' *Autonomous Robots*, vol. 33, pp. 1:173-188, 2012.

[4] Bachrach, A., He, R., Roy, N., 'Autonomous Flight in Unknown Indoor Environment,' *Int'l Journal of Micro Air Vehicles*, vol. 1, pp. 4:217-228, 2009.

[5] Bachrach, A., Prentice, S., He, R., Roy, N., 'RANGE - Robust Autonomous Navigation in GPS-denied Environments,' *Journal of Field Robotics*, vol. 28, pp. 5:644-666, 2011.

[6] Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A., Krainin, M., Maturana, D., Fox, D., Roy, N.,'Estimation, Planning and Mapping for Autonomous Flight Using an RGB-D Camera in GPS-denied Environments,' *International Journal of Robotics Research*, vol. 31, pp. 11:1320-1343, 2012.

[7] Weiss, S., Scaramuzza, D., Siegwart, R., 'Monocular-SLAM-Based Navigation for Autonomous Micro Helicopters in GPS-denied Environments,' *Journal of Field Robotics*, vol. 28, pp. 6:854-874, 2011.

[8] Weiss, S., Achtelik, M. W., Lynen, S., Achtelik, M. C., Kneip, L., Chli, M., Siegwart, R., 'Monocular Vision for Long-term Micro Aerial VehicleState Estimation: A Compendium,' *Journal of Field Robotics*, vol. 30, pp. 5:803-831, 2013.

[9] Teulière, C., Eck, L., Marchand, E., 'Chasing a Moving Target from a Flying UAV,' in *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sept. 2011, pp. 4929-4934.

[10] Nakamura, T., Haviland, S., Bershadsky, D., Magree, D., Johnson, E., 'Vision-Based Closed-Loop Tracking Using Micro Air Vehicles,' *IEEE Aerospace Conference*, pp. 1-12, 2016.

[11] Gomez-Balderas, J. E., Flores, G., Carrillo, L. R., Lozano R.,'Tracking a Ground Moving Target with a Quadrotor Using Switching Control', *Journal of Intelligent and Robotic System*, pp. 70:65-78, 2013.

[12] Bohdanov, D., *Quadrotor UAV Control for Vision-Based Moving Target Tracking Task*. MS Thesis. University of Toronto, 2012.

[13] Lee, D., Ryan, T., Kim, H., 'Autonomous Landing of a VTOL UAV on a Moving Platform Using Image-Based Visual Servoing,' in *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN, May 2012, pp. 971-976.

[14] B. Heriss, T. Hamel, R. Mahony and F. X. Russotto, 'Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow,' in *IEEE Trans. on Robotics*, vol. 28, no. 1, pp. 77-89, Feb. 2012.

[15] P. Serra; R. Cunha; T. Hamel; D. Cabecinhas; C. Silvestre, 'Landing of a Quadrotor on a Moving Target Using Dynamic Image-Based Visual Servo Control,' in *IEEE Trans. on Robotics*, vol.PP, pp.1-12, Dec. 2016.

[16] Dobrokhodov, V., Kaminer, I., Jones, K., Ghabcheloo, R., 'Vision-Based Tracking and Motion Estimation for Moving Targets Using Unmanned Air Vehicles,' *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 4:907-917, July 2008.

[17] Zhang, M., Liu, H., 'Vision-Based Tracking and Estimation of Ground Moving Target Using Unmanned Aerial Vehicle,' in *Proc. of 2010 American Control Conf.*, Baltimore, MD, June 2010, pp. 6968-6973.

[18] Ma, Y., Soatto, S., Košecká, J., Sastry, S. S., An Invitation to 3D Vision: From Images to Geometric Models, Springer, 2004, pp. 48-58.

[19] Bernstein, D., Geometry, Kinematics, Statics, and Dynamics, Department of Aerospace Engineering, University of Michigan, 2015, p. 118.

[20] Khalil, H., Nonlinear Systems, Prentice Hall, 2002, pp. 346-350.

[21] De Nardi, R., 'The QRSim Quadrotors Simulator', *Research Note RN/13/08*, Dept. of Computer Science University, College London, March 2013.

[22] Csurka, G., Dance, C.R,, Fan, L., Willamowski, J., Cedric, B., 'Visual categorization with bags of keypoints,' *Statistical Learning in Computer Vision, in European Conf. on Computer Vision*, pp. 1-22, 2004.

[23] Jegou, H., Douze, M., Schmid, C., Perez, P., 'Aggregating local descriptors into a compact image representation,' *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3304-3311, 2010.

[24] Perronnin, F., Sanchez, J., Mensink, T., 'Improving the fisher kernel for large-scale image classification,' *European Conf. on Computer Vision*, pp. 143-156, 2010.

[25] Lazebnik, S., Schmid, C., Ponce, J., 'Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,' *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2:2169-2178, 2006.

[26] He, K., Zhang, X., Ren, S., Sun, J., 'Spatial pyramid pooling in deep convolutional networks for visual recognition,' *European Conf. on Computer Vision*, pp. 346-361, 2014.

[27] Zeiler, M.D, Fergus, R., 'Visualizing and understanding convolutional networks,' *European Conf. on Computer Vision*, pp. 818-833, 2014.

[28] Manen, S., Guillaumin, M., Van Gool, L., 'Prime object proposals with randomized Prim's algorithm,' *IEEE Int'l Conf. on Computer Vision*, pp. 2536-2543, 2013.

[29] Alexe, B., Deselaers, T., Ferrari, V., 'What is an object?,' *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 73-80, 2010.

[30] Lowe, D.G., 'Object recognition from local scale-invariant features,' *IEEE Int'l Conf. on Computer Vision*, pp. 2:1150-1157, 1999.

[31] Arthur, D., Vassilvitskii, S., 'k-means++: The advantages of careful seeding,' *ACM-SIAM Symp. on Discrete algorithms*, pp. 1027-1035, 2007.

[32] Bishop, C.M., Pattern Recognition and Machine Learning, Springer, 2007, pp. 97-105.