2017-09-19

Relative target estimation using a cascade of extended Kalman filters

Jerel Nielsen
jerel.nielsen@gmail.com

Randal Beard
Brigham Young University - Provo, beard@byu.edu

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

Relative target estimation using a cascade of extended Kalman filters

Jerel Nielsen, *Brigham Young University*
Randal W. Beard, *Brigham Young University*

BIOGRAPHY

J. Nielsen is a graduate student in the Multiple AGent Intelligent Coordination and Control (MAGICC) laboratory at BYU working toward a Ph.D. in Electrical Engineering. R. Beard is a Professor in the Electrical Engineering Department at Brigham Young University.

ABSTRACT

This paper presents a method of tracking multiple ground targets from an unmanned aerial vehicle (UAV) in a 3D reference frame. The tracking method uses a monocular camera and makes no assumptions on the shape of the terrain or the target motion. The UAV runs two cascaded estimators. The first is an Extended Kalman Filter (EKF), which is responsible for tracking the UAV's state, such as position and velocity relative to a fixed frame. The second estimator is an EKF that is responsible for estimating a fixed number of landmarks within the camera's field of view. Landmarks are parameterized by a quaternion associated with bearing from the camera's optical axis and an inverse distance parameter. The bearing quaternion allows for a minimal representation of each landmark's direction and distance, a filter with no singularities, and a fast update rate due to few trigonometric functions. Three methods for estimating the ground target positions are demonstrated: the first uses the landmark estimator directly on the targets, the second computes the target depth with a weighted average of converged landmark depths, and the third extends the target's measured bearing vector to intersect a ground plane approximated from the landmark estimates. Simulation results show that the third target estimation method yields the most accurate results.

INTRODUCTION

Autonomous multiple target tracking (MTT) has been an area of interest for several decades, due to a broad list of applications, such as law enforcement [1], air traffic control [2], collision avoidance [3], simultaneous localization and mapping (SLAM) [4], tracking space debris [5], and others. There are typically three main difficulties in MTT: collecting sensor measurements of the targets, associating these measurements with the correct targets (data association), and filtering clutter or noise from true target measurements. Measurement collection is computationally expensive when image processing is required, for example, when tracking with a video camera. The difficulty of data association grows with the number of targets and the amount of clutter in the measurements.

Many algorithms have been developed to solve the MTT problem, such as the multiple hypothesis tracking (MHT) [6], joint probabilistic data association (JPDA) [7], and probabilistic hypothesis density (PHD) [8] algorithms. These suffer from computational complexity, requirements of prior information, poor track continuity, or large variance in target estimates. Recursive RANSAC (R-RANSAC) is a recently introduced MTT algorithm [9, 10, 11, 12, 13] that solves many of these issues. R-RANSAC extends the traditional random sample consensus (RANSAC) algorithm to recursively estimate multiple dynamic signals in clutter. It stores a set of track hypotheses and identifies the best hypothesis of each target's track, and given a sliding window of measurements, it either updates the existing hypotheses with a Kalman update or generates new tracks with the set of measurements using RANSAC. This algorithm can run in real time, even with a large number of targets and a significant amount of clutter.

For video-based tracking, the algorithm can operate directly in the undistorted, 2D image frame [12], or it can operate in the fixed, 3D fixed frame [14]. In order to track in the fixed frame, measurements are first projected from image coordinates into the camera frame via a perspective projection, followed by several rotations and translations to the fixed frame. This process and

coordinate systems used in this paper are described in detail in the appendix. Due to ease of implementation, MTT is typically done in the image frame using a nearly constant velocity, acceleration, or jerk model for motion propagation. Advantages of object tracking in the image frame include noise introduced into the measurement only comes from one sensor (the camera), image noise is typically minimal for modern cameras, and the tracker does not depend on the quality of the camera pose estimator. Advantages of fixed-frame tracking include the ability to use target specific motion propagation models, tracks are readily available for multi-vehicle problems, the tracker can account for geography, and poor homographies do not directly affect tracking. Calculation of the relative ground position of a target from a UAV is commonly done via the flat-earth model [15], which assumes the UAV flies above a flat plane. This will clearly experience large errors in real scenarios where the ground is uneven or the UAV altitude is not accurately known. The methods presented in this paper are capable of 3D target localization without this flat-earth assumption.

The primary contribution proposed in this paper is a novel method of estimating 3D target positions relative to the camera. Three methods to overcome the flat-earth assumption are presented and simulation results are discussed. Each method relies on a landmark estimator, which uses concepts discussed in the recently introduced robust visual-inertial odometry (ROVIO) [16] algorithm. The first method is to directly insert bearing vector measurements of targets into the landmark estimator with additional process noise added. The second estimates the target's distance as a weighted average of the landmark estimator's converged landmark distances. The third method approximates a ground plane via least squares, given three or more converged landmark estimates, where the target distance is estimated as the intersection of its bearing vector and the approximate ground plane.

The data pipeline is shown in Figure 1. The UAV State Estimator is an EKF responsible for estimating the position and velocity of the UAV relative to a fixed, local Euclidean reference frame parameterized in North-East-Down (NED) coordinates. This estimator also allows us to compute the camera linear and angular velocities, which are needed inputs to the Landmark Estimator. The Landmark Estimator filters out the moving image measurements and Target Estimator filters out the static image measurements. As the Landmark Estimator's estimates of static landmarks converge, the Target Estimator produces moving target measurements in the 3D camera frame. These can then be transformed into the fixed frame, where they are then processed by a multiple target tracker (R-RANSAC) to produce track estimates also in the fixed frame. The main focus of this paper is the Target Estimator block, however, the Landmark Estimator is also outlined and details of its derivation is contained in the appendix.

LANDMARK ESTIMATION

Landmarks are objects that are static relative to the fixed frame and are detected by a camera as features or corners. These features are direct measurements of landmark direction and can be used for tightly coupled visual-inertial odometry [16]. In this paper, we assume that the UAV has a state estimator separate from a landmark estimator. The landmark estimator is a multiplicative extended Kalman filter (MEKF) that estimates the bearing vector and inverse distance to each landmark, given camera linear/angular velocity and landmark measurements. Camera velocities come from the state estimator and are used to propagate the state forward in time. The landmark bearing measurements update the landmark estimates.

Propagation

The landmark state is comprised of a set of quaternions associated with the bearing vectors of N landmarks and also the inverse distances to those same landmarks, and is given by

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_0^\top & \rho_0 & \cdots & \mathbf{q}_N^\top & \rho_N \end{bmatrix}^\top, \tag{1}$$

where $\mathbf{q}_i \in SO(3)$ is the shortest rotation from the camera optical axis to the landmark bearing vector represented by a unit quaternion and $\rho_i \in \mathbb{R}$ is the inverse depth parameter associated with the i^{th} landmark. All landmarks are assumed to be static and have the same kinematics defined by

$$\dot{\mathbf{q}} = f_q(\boldsymbol{\omega}, \mathbf{v}) = T_\zeta^\top \left(\boldsymbol{\omega}_{c/i}^c + \rho \left(\zeta^c\right)^\times \mathbf{v}_{c/i}^c \right) \tag{2}$$

$$\dot{\rho} = f_\rho(\mathbf{v}) = \rho^2 \left(\zeta^c\right)^\top \mathbf{v}_{c/i}^c. \tag{3}$$

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

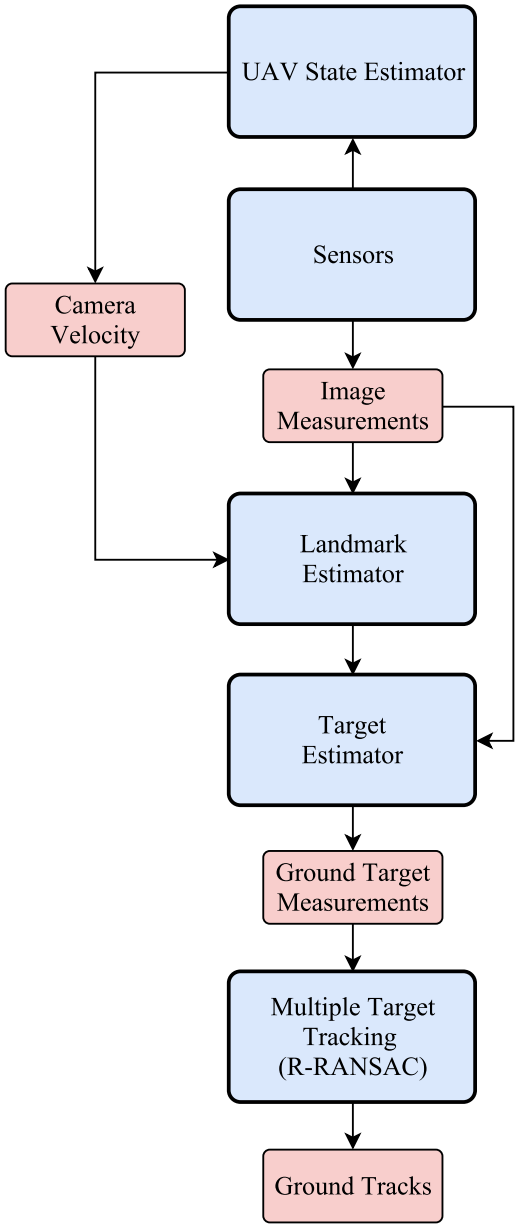

Figure 1: The data pipeline for tracking multiple ground targets in the fixed frame. Blue blocks represent functions and the red blocks represent outputs of these functions.

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

where $\mathbf{v}^c_{c/i}$ and $\omega^c_{c/i}$ are the linear and angular velocities of the camera w.r.t. the fixed frame, $\zeta^c \in S^2$ is the unit vector pointing at the landmark, defined in the camera frame, and T_ζ is a projection discussed in the Conventions section of the appendix.

We use a quaternion as the underlying representation for the bearing to each landmark, thus, the state has $5N$ dimensions, while the kinematics of the state has $3N$ dimensions. This difference in dimensions is because elements of $S^2 \subset \mathbb{R}^3$ are minimally represented with two parameters (typically azimuth and elevation) and in order to keep the covariance minimal, changes to the state must be performed in a minimal manner.

The vector and quaternion components of the state, that is the inverse depth and quaternion components, can be modified on each one's particular manifold [17]. Euler integration of the kinematics is used to propagate the state forward in time. Propagation of the unit quaternions representing landmark unit vectors [18] and inverse depth parameters for a small time step Δt is given by

$$\hat{\mathbf{q}}(t+\Delta t) = \hat{\mathbf{q}}(t) \boxplus \hat{T}_\zeta \left(f_q(\omega, \mathbf{v}) \Delta t \right) \tag{4}$$

$$\hat{\rho}(t+\Delta t) = \hat{\rho}(t) + f_\rho(\mathbf{v}) \Delta t, \tag{5}$$

where the $\boxplus$/$\boxminus$ operators are discussed in the Conventions section of the appendix.

The change in state covariance w.r.t. time is given by

$$\dot{P} = FP + PF^\top + GQ_uG^\top + Q_x, \tag{6}$$

where the Jacobians F and G are derived in the appendix. The state covariance is then also propagated forward with Euler integration by

$$P(t+\Delta t) = P(t) + \left(FP + PF^\top + GQ_uG^\top + Q_x \right) \Delta t. \tag{7}$$

Update

The measurement residual error for a general measurement with measurement model $\mathbf{h}$ is computed by

$$\mathbf{r} = \mathbf{z} \boxminus \mathbf{h}(\hat{\mathbf{x}}), \tag{8}$$

where $\mathbf{z}$ is a sensor measurement.

The measurement uncertainty and Kalman gain are computed by

$$S = HPH^\top + R_\mathbf{z} \tag{9}$$

$$K = PH^\top S^{-1}. \tag{10}$$

The update is usually given by

$$\mathbf{x}^+ = \mathbf{x}^- + K\mathbf{r}, \tag{11}$$

but the quaternion components cannot be correctly updated with addition. Separating $K\mathbf{r}$ into a vector state $\Delta\rho$ associated with the inverse depth components and an attitude state $\Delta\mathbf{q}$ associated with the bearing quaternions, we perform the update by

$$\hat{\mathbf{q}}^+ = \hat{\mathbf{q}}^- \boxplus T^-_\zeta \Delta\mathbf{q} \tag{12}$$

$$\hat{\rho}^+ = \hat{\rho}^- + \Delta\rho. \tag{13}$$

Lastly, the covariance is updated by

$$P^+ = (I - KH) P^-. \tag{14}$$

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

Measurement model

The landmark bearing vectors are measured directly in the camera frame, so the measurement model is given by

$$\mathbf{h}(\hat{\mathbf{x}}) = \mathbf{q} \boxplus \eta, \tag{15}$$

where η is Gaussian noise and the measurement Jacobian is then given by

$$H = \frac{\partial \mathbf{h}(\hat{\mathbf{x}})}{\partial \mathbf{x}} \tag{16}$$

$$= \begin{bmatrix} I & \mathbf{0} \end{bmatrix}. \tag{17}$$

TARGET ESTIMATION METHODS

The first method for estimating a moving target's 3D position relative to the UAV is to use the static landmark estimator directly. The problem with this method is that the kinematics of the estimator are derived from static landmark assumptions. Therefore, this works for slow moving targets to some degree by increasing the process noise parameters of the estimator, but as shown in the Results section, it is far from ideal.

The second method requires at least one landmark's inverse distance state to converge. Assuming that this and other landmarks are near the target, we can take a weighted average of the landmarks' distances to get the target's estimated distance. The weights come from the inverse error in the target's measured bearing vector against the landmarks' estimated bearing vectors. This means that landmark bearing vectors closer to the target's have more influence in the calculation of the target's distance. Suppose we have a target's bearing vector ζ_t and several landmark vectors $\zeta_0 \cdots \zeta_N$. The target's inverse distance is estimated as

$$\hat{\rho}_t = \frac{\sum_{i=0}^{N} w_i \rho_i}{\sum_{i=0}^{N} w_i}, \qquad w_i = \frac{1}{\|\zeta_t - \zeta_i\|}. \tag{18}$$

In the third target estimation method, we fit a ground plane to three or more landmarks whose inverse distance estimates have converged below some threshold and then find the distance required to scale the target's bearing vector to intersect this approximated ground plane. It's important to note that this is done in the camera reference frame. The equation of a plane is given by

$$\mathbf{n}^\top (\mathbf{r} - \mathbf{r}_0) = 0, \tag{19}$$

where $\mathbf{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^\top$ is the vector normal to the plane, $\mathbf{r} = \begin{bmatrix} x & y & z \end{bmatrix}^\top$ is an arbitrary point on the plane, and $\mathbf{r}_0 = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^\top \delta$ is the offset vector to the center of the plane. Evaluating this dot product and rearranging variables yields

$$\frac{n_x}{n_z} x + \frac{n_y}{n_z} y + z = \frac{n_x}{n_z} x_0 + \frac{n_y}{n_z} y_0 + z_0, \tag{20}$$

and if we let $a = \frac{n_x}{n_z}$, $b = \frac{n_y}{n_z}$, and $c = \frac{n_x}{n_z} x_0 + \frac{n_y}{n_z} y_0 + z_0$, we then have

$$ax + by + z = c. \tag{21}$$

Given $N \geq 3$ points, a plane can be approximated by a least squares solution of the form

$$A\mathbf{x} = \mathbf{b}, \tag{22}$$

where

$$A = \begin{bmatrix} x_1 & y_1 & -1 \\ x_2 & y_2 & -1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & -1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -z_1 \\ -z_2 \\ \vdots \\ -z_N \end{bmatrix}, \tag{23}$$

and where the solution is $\mathbf{x} = \left(A^\top A\right)^{-1} A^\top \mathbf{b}$. With the ground plane defined and assuming that the target bearing vector is not parallel to this plane, we can find the distance $d(\rho_t)$ which scales the target bearing vector $\zeta_t = \begin{bmatrix}\zeta_x & \zeta_y & \zeta_z\end{bmatrix}^\top$ to intersect this ground plane. Inserting $\mathbf{r} = d(\rho_t)\,\zeta_t$ into the general equation of a plane yields

$$\mathbf{n}^\top \left(d(\rho_t)\,\zeta_t - \mathbf{r}_0\right) = 0, \tag{24}$$

and performing the dot product and some algebra gives

$$d(\rho_t)\left(\frac{n_x}{n_z}\zeta_x + \frac{n_y}{n_z}\zeta_y + \zeta_z\right) = \frac{n_x}{n_z}x_0 + \frac{n_y}{n_z}y_0 + z_0. \tag{25}$$

Once again, letting $a = \frac{n_x}{n_z}$, $b = \frac{n_y}{n_z}$, and $c = \frac{n_x}{n_z}x_0 + \frac{n_y}{n_z}y_0 + z_0$, and solving for $d(\rho_t)$ leaves us with

$$d(\rho_t) = \frac{c}{a\zeta_x + b\zeta_y + \zeta_z}. \tag{26}$$

SIMULATION RESULTS

Simulations were used to demonstrate each of the proposed methods for relative 3D target position estimation. A quadcopter flies to user-specified points in a 50 by 50 meter square, 50 meters above the ground. Landmarks are placed with a uniformly random distribution inside of this square area and targets are simply points moving in figure eight patterns at various frequencies. Quadcopter kinematics have become commonplace in the literature, so they are not detailed here but are used in simulation to generate UAV position and velocities. The landmark estimator receives linear and angular velocities of the quadcopter from the UAV State Estimator as shown in Figure 1. Since the focus of this paper is not the UAV state estimator, we assume that it is perfect but then corrupt the measurements of landmarks and targets with Gaussian noise.

Because the bearing to each landmark and target is measured by the camera, it is considered known. Distance or depth, however, is not known and is recursively estimated by the Landmark Estimator, as shown in Figure 2. This shows a single landmark estimate over time as an example but is typical of most landmark estimates, where the distance estimates converge in about a second with very little uncertainty.

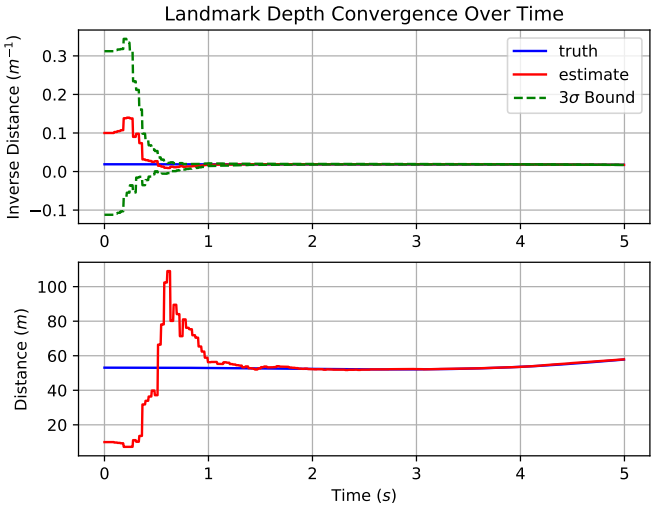

Figure 2: An example of the convergence of stationary landmarks over time.

Results of the first target estimation method are shown in Figures 3 and 4. In Figure 3, we see that the uncertainty only shrinks to a certain point and the distance error is as much as 20 meters at times. The error in distance translates directly to the estimated north, east, and down components of the target shown in Figure 4, where we get errors of ten meters or more on each axis. This

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

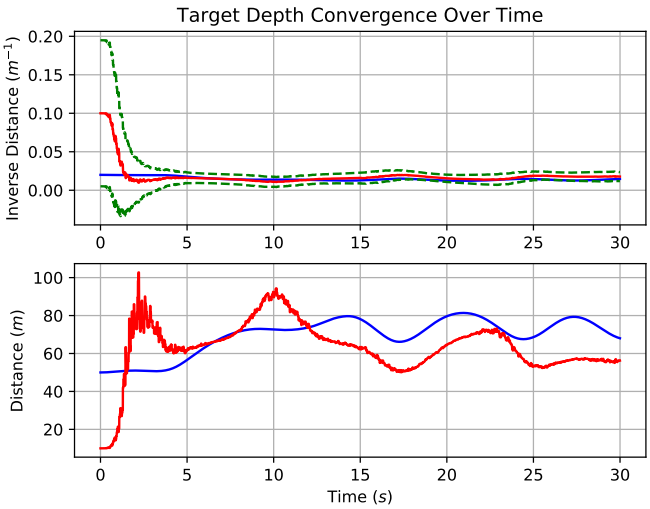

Figure 3: An example of the convergence of a moving target tracked directly in the landmark estimator over time.

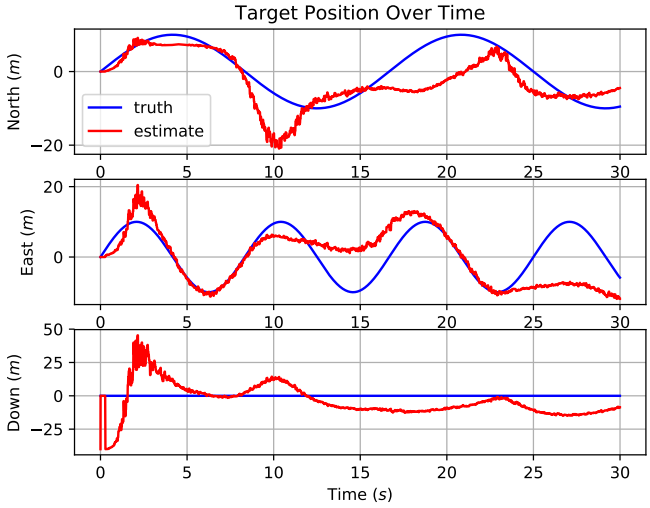

Figure 4: Tracking positions of a target in the fixed frame over time using the filter directly. North, east, and down components are shown.

is expected because the kinematic model is derived for a stationary landmark. However, this method of target estimation may work for very slow moving targets with a sufficiently high camera frame rate.

The second target estimation method was used in Figure 5. We see that using the weighted average of static landmark errors to estimate the 3D target position works better than the first method. The estimates of both targets have error as high as five meters but roughly exhibit the correct trends. The estimate shows up as zero for the initial quarter of a second because it waits for at least one landmark estimate to converge.

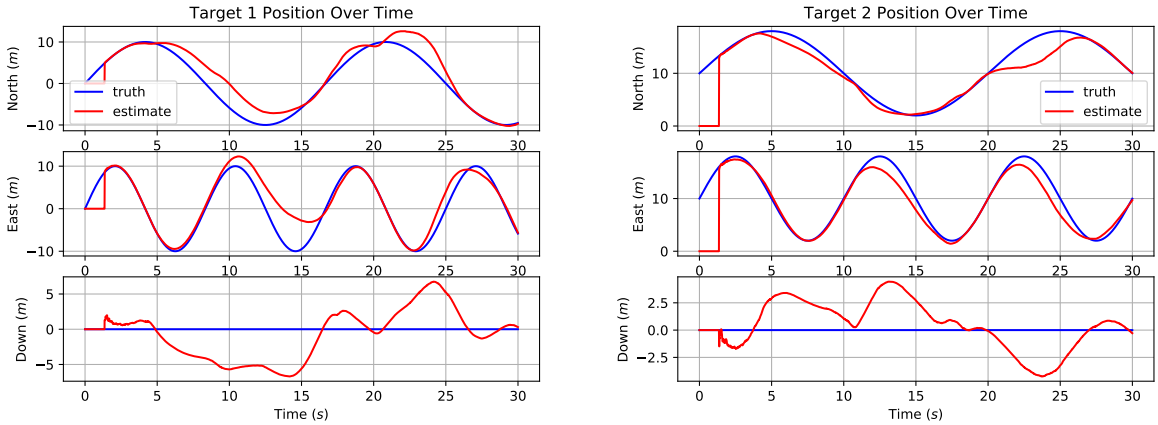

Figure 5: Tracking positions of two targets in the fixed frame over time using the average bearing method. North, east, and down components are shown.

The third method of target estimation, using ground plane fitting, is clearly the best, as shown in Figure 6. We see that the error is around a meter or less on all axes for both targets being tracked. Notice that the north and east components are estimated almost exactly. This is because the quadrotor is high above the targets, so these components are more affected by the bearing vector estimates of landmarks, which are easily estimated. The down component is subject to the distance estimates of landmarks, which is less precise than the bearings. Thus, the fitted ground plane tends to bounce up and down, causing the down estimate to wander.

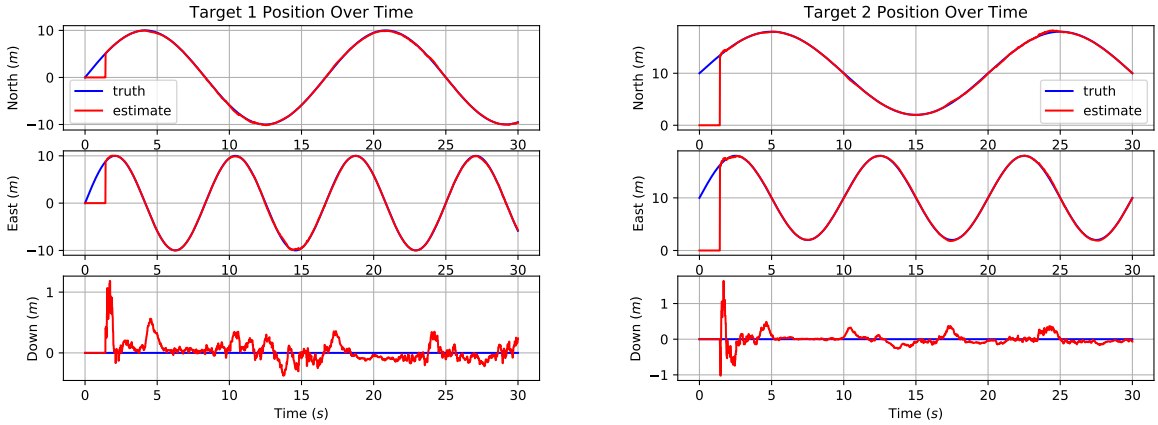

Figure 6: Tracking positions of two targets in the fixed frame over time using the fitted plane method. North, east, and down components are shown.

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

CONCLUSIONS

We have developed a method for estimating 3D ground target positions from a UAV equipped with a monocular camera. It is clear that fitting a plane to estimates of landmarks on the ground is the best of the discussed methods. It assumes that the UAV state estimator is working properly and that most of the estimated landmarks are on the ground. Future work includes adding a constant velocity model to the landmark estimator state to track moving targets directly with better accuracy, incorporating RANSAC in the ground plane fitting to eliminate landmarks such as trees or other tall objects, and running this algorithm in hardware on aerial platforms. Accuracy in a real tracking scenario would also be improved by selecting a region of interest around each target and creating a planar approximation based on landmark estimates nearby each target.

APPENDIX

This appendix contains useful derivations related to the landmark estimator, some of which can also be found in [18] using a different notation.

Nomenclature

R_a^b	Rotation from reference frame a to b
$\hat{a}$	Estimate of true variable a
$\bar{a}$	Measurement of variable a
$\dot{a}$	Time derivative variable a
$\tilde{a}$	Error of variable a, i.e., $\tilde{a} = a - \hat{a}$

Superscript

i	Expressed in the fixed or inertial coordinate frame
b	Expressed in the vehicle body coordinate frame
g	Expressed in the gimbal coordinate frame
cb	Expressed in the camera body coordinate frame
c	Expressed in the camera coordinate frame
l	Expressed in the landmark coordinate frame
p	Expressed in the image coordinate frame
$\top$	Matrix transpose

Subscript

a/b	Velocity or angular rate of frame a w.r.t. frame b
ab	Vector from a to b

Conventions

This section defines key mathematical operations used throughout the paper. Quaternions take the form

$$\mathbf{q} = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k} = \begin{bmatrix} q_w & \bar{\mathbf{q}}^\top \end{bmatrix}^\top, \tag{27}$$

where $\bar{\mathbf{q}} = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^\top$ and the standard Hamiltonian convention is used, which is defined by

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1. \tag{28}$$

Quaternion multiplication is defined by

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w & -\bar{\mathbf{p}}^\top \\ \bar{\mathbf{p}} & p_w I + \bar{\mathbf{p}}^\times \end{bmatrix} \begin{bmatrix} q_w \\ \bar{\mathbf{q}} \end{bmatrix} \tag{29}$$

$$= \begin{bmatrix} q_w & -\bar{\mathbf{q}}^\top \\ \bar{\mathbf{q}} & q_w I - \bar{\mathbf{q}}^\times \end{bmatrix} \begin{bmatrix} p_w \\ \bar{\mathbf{p}} \end{bmatrix}, \tag{30}$$

and the operator $(\cdot)^\times$ is the skew-symmetric operator defined by

$$\mathbf{a}^\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}. \tag{31}$$

A 3×3 rotation matrix associated with $\mathbf{q}_a^b$ may be defined as

$$R_a^b = R\left(\mathbf{q}_a^b\right) = \left(2q_w^2 - 1\right)\mathbf{I} - 2q_w\bar{\mathbf{q}}^\times + 2\bar{\mathbf{q}}\bar{\mathbf{q}}^\top. \tag{32}$$

Landmarks in this paper move w.r.t. the camera, similar to traditional, inertial kinematics, and therefore, the following definitions are used

$$\mathbf{q} \triangleq \hat{\mathbf{q}} \otimes \tilde{\mathbf{q}} \tag{33}$$

$$\tilde{\mathbf{q}} = \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \tag{34}$$

$$R(\mathbf{q}) = R(\tilde{\mathbf{q}})R(\hat{\mathbf{q}}). \tag{35}$$

We use the same representation of bearing vectors to landmarks as [18]. The bearing vector and its tangent space operator are given by

$$\zeta = R(\mathbf{q})\,\mathbf{e}_z \in S^2 \subset \mathbb{R}^3 \tag{36}$$

$$T_\zeta = R(\mathbf{q})\begin{bmatrix}\mathbf{e}_x & \mathbf{e}_y\end{bmatrix} \in \mathbb{R}^{3\times 2}, \tag{37}$$

where $\mathbf{e}_{x/y/z} \in \mathbb{R}^3$ are basis vectors of an arbitrary, orthonormal coordinate system and the unit quaternion $\mathbf{q}$ is defined by the shortest rotation from the camera optical axis to the bearing vector ζ. Note that we now have two different quaternions for attitude and for a bearing vector, where the attitude quaternion has a superscript and subscript for the coordinate systems and the bearing vector quaternion has no coordinate system superscript or subscript. The bearing vector is over-parameterized because it is a three element, unit length vector, therefore only two parameters (such as azimuth and elevation) are needed to fully describe it. This leads to a singularity in the covariance associated with ζ, so we will remove one element by mapping to the space orthogonal to ζ. The matrix T_ζ spans this tangent space, where $T_\zeta : \mathbb{R}^2 \to \mathbb{R}^3$ and $T_\zeta^\top : \mathbb{R}^3 \to \mathbb{R}^2$. In other words, $T_\zeta^\top$ maps to the 2-dimensional tangent space of ζ spanned by $\mathbf{e}_x$ and $\mathbf{e}_y$.

The landmark estimator uses boxplus and boxminus operators [17], which were applied to unit vector kinematics in [18]. In this paper, they are defined slightly differently than [18] by

$$\boxplus : SO(3) \times \mathbb{R}^2 \to SO(3), \tag{38}$$

$$\mathbf{q}, \tilde{\mathbf{q}} \mapsto \mathbf{q} \otimes \exp\left(T_\zeta \tilde{\mathbf{q}}\right) \tag{39}$$

$$\boxminus : SO(3) \times SO(3) \to \mathbb{R}^2 \tag{40}$$

$$\mathbf{p}, \mathbf{q} \mapsto T_\zeta^\top \theta(\mathbf{q}, \mathbf{p}), \tag{41}$$

where

$$\theta(\mathbf{q}, \mathbf{p}) = \cos^{-1}\left(\zeta_p^\top \zeta_q\right) \frac{\zeta_p \times \zeta_q}{\left\|\zeta_p \times \zeta_q\right\|}. \tag{42}$$

Given an angular update $\delta \in \mathbb{R}^3$, the exponential map to a unit quaternion is given by

$$\exp(\delta) = \begin{bmatrix} q_w \\ \bar{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\|\delta\|}{2}\right) \\ \sin\left(\frac{\|\delta\|}{2}\right)\frac{\delta}{\|\delta\|} \end{bmatrix}, \tag{43}$$

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

and when $\|\delta\| \approx 0$,

$$\exp(\delta) \approx \begin{bmatrix} 1 \\ \frac{\delta}{2} \end{bmatrix}. \tag{44}$$

The corresponding logarithm is given by

$$\log(\mathbf{q}) = 2\text{atan2}\left(\|\bar{\mathbf{q}}\|, q_w\right) \frac{\bar{\mathbf{q}}}{\|\bar{\mathbf{q}}\|}, \tag{45}$$

and when $\|\bar{\mathbf{q}}\| \approx 0$,

$$\log(\mathbf{q}) \approx \text{sign}(q_w)\, \bar{\mathbf{q}}. \tag{46}$$

Coordinate reference frames are similar to those defined in [15]. There is a fixed, inertial coordinate system. The vehicle frame is centered on the vehicle body with its axes aligned with the inertial frame. The body frame is the vehicle frame rotated so that its axes align with the body. The gimbal frame is the body frame translated to the gimbal and rotated to align with the gimbal. The camera body frame is the gimbal frame translated and rotated to align with the camera. Typically, camera coordinates have the z-axis aligned with the camera's optical axis and x-axis to the right. Therefore, the rotation from the camera body frame to the camera frame is a simple reordering of the axes. Finally, from the camera frame, we can perform a perspective projection into the image frame via the intrinsic camera matrix, assuming scene depth is known. We can also rotate from the camera frame to the landmark frame, which aligns the optical axis with a landmark bearing vector via shortest axis-angle rotation.

Common identities and key derivatives

Here are some common identities that prove useful in these derivations

$$(R\mathbf{v})^\times = R\mathbf{v}^\times R^\top \tag{47}$$

$$\dot{R}\left(\mathbf{q}_b^i\right) = R\left(\mathbf{q}_b^i\right)\left(\boldsymbol{\omega}_{b/i}^b\right)^\times \tag{48}$$

$$\dot{R}\left(\mathbf{q}_i^b\right) = -\left(\boldsymbol{\omega}_{b/i}^b\right)^\times R\left(\mathbf{q}_i^b\right). \tag{49}$$

The following derivatives are taken from [18],

$$\frac{\partial}{\partial t}\mathbf{q}_i^b = -\boldsymbol{\omega}_{i/b}^b \tag{50}$$

$$\frac{\partial}{\partial \mathbf{q}}\left(R(\mathbf{q})\,\mathbf{r}\right) = -\left(R(\mathbf{q})\,\mathbf{r}\right)^\times \tag{51}$$

$$\frac{\partial}{\partial \mathbf{q}}\zeta^c = \left(\zeta^c\right)^\times T_\zeta \tag{52}$$

$$\frac{\partial}{\partial \mathbf{q}}\left(T_\zeta^\top \mathbf{r}\right) = -T_\zeta^\top (\mathbf{r})^\times T_\zeta. \tag{53}$$

Time derivative of feature distance

The distance to a feature in the camera frame is given by

$$d(\rho) = \frac{1}{\rho}. \tag{54}$$

Differentiating w.r.t. time yields

$$\frac{d}{dt}\left(d(\rho)\right) = \frac{d}{d\rho}\left(d(\rho)\right)\frac{d}{dt}(\rho) \tag{55}$$

$$= d'(\rho)\,\dot{\rho}, \tag{56}$$

where

$$d'(\rho) = \frac{d}{d\rho}(d(\rho)) \tag{57}$$
$$= \frac{d}{d\rho}\left(\frac{1}{\rho}\right) \tag{58}$$
$$= -\frac{1}{\rho^2}. \tag{59}$$

Time derivative of a bearing vector

A unit vectors pointing to a feature in the camera frame is over-parameterized by a quaternion representation of $SO(3)$ or a vector in $\mathbb{R}^3$, since its minimal representation is in S^2, the unit sphere in $\mathbb{R}^3$. Therefore, its derivative should be represented in a minimal manner. As discussed in [18], the change in a unit vector can be described by an axis-angle rotation vector, $\dot{\zeta} \in \mathbb{R}^2$, in the plane orthogonal to the unit vector. Thus, the time derivative of a unit vector may be given by

$$\frac{\partial}{\partial t}(\zeta^c) = \frac{\partial}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial t}(\zeta^c) \tag{60}$$
$$= \frac{\partial}{\partial \mathbf{q}}(\zeta^c)\frac{\partial \mathbf{q}}{\partial t} \tag{61}$$
$$= (\zeta^c)^{\times} T_\zeta \dot{\zeta}^c. \tag{62}$$

It's important to note that $\dot{\zeta}^c$ indicates a rotation from the new unit vector direction to the old, and thus, the above equation creates a vector pointing from the old direction to the new.

Bearing vector tangent space mapping identities

Mapping from the unit vector tangent space and immediately back gives

$$T_\zeta^\top T_\zeta = \left(R(\mathbf{q})\begin{bmatrix}\mathbf{e}_x & \mathbf{e}_y\end{bmatrix}\right)^\top R(\mathbf{q})\begin{bmatrix}\mathbf{e}_x & \mathbf{e}_y\end{bmatrix} \tag{63}$$
$$= \begin{bmatrix}\mathbf{e}_x^\top \\ \mathbf{e}_y^\top\end{bmatrix} R^\top(\mathbf{q}) R(\mathbf{q}) \begin{bmatrix}\mathbf{e}_x & \mathbf{e}_y\end{bmatrix} \tag{64}$$
$$= \begin{bmatrix}\mathbf{e}_x^\top \\ \mathbf{e}_y^\top\end{bmatrix} \begin{bmatrix}\mathbf{e}_x & \mathbf{e}_y\end{bmatrix} \tag{65}$$
$$= \begin{bmatrix}\mathbf{e}_x^\top \mathbf{e}_x & \mathbf{e}_x^\top \mathbf{e}_y \\ \mathbf{e}_y^\top \mathbf{e}_x & \mathbf{e}_y^\top \mathbf{e}_y\end{bmatrix} \tag{66}$$
$$= \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} \tag{67}$$
$$= I. \tag{68}$$

Published in the proceedings of ION GNSS+ 2017.
https://www.ion.org/publications/abstract.cfm?articleID=15158

Reversing the transformation order, we now have

$$T_\zeta T_\zeta^\top = R(\mathbf{q}) \begin{bmatrix} \mathbf{e}_x & \mathbf{e}_y \end{bmatrix} \left(R(\mathbf{q}) \begin{bmatrix} \mathbf{e}_x & \mathbf{e}_y \end{bmatrix} \right)^\top \tag{69}$$

$$= R(\mathbf{q}) \begin{bmatrix} \mathbf{e}_x & \mathbf{e}_y \end{bmatrix} \begin{bmatrix} \mathbf{e}_x^\top \\ \mathbf{e}_y^\top \end{bmatrix} R^\top(\mathbf{q}) \tag{70}$$

$$= R(\mathbf{q}) \left(\mathbf{e}_x \mathbf{e}_x^\top + \mathbf{e}_y \mathbf{e}_y^\top \right) R^\top(\mathbf{q}) \tag{71}$$

$$= R(\mathbf{q}) \left(\mathbf{e}_x \mathbf{e}_x^\top + \mathbf{e}_y \mathbf{e}_y^\top + \mathbf{e}_z \mathbf{e}_z^\top - \mathbf{e}_z \mathbf{e}_z^\top \right) R^\top(\mathbf{q}) \tag{72}$$

$$= R(\mathbf{q}) \left(I - \mathbf{e}_z \mathbf{e}_z^\top \right) R^\top(\mathbf{q}) \tag{73}$$

$$= R(\mathbf{q}) R^\top(\mathbf{q}) - R(\mathbf{q}) \mathbf{e}_z \mathbf{e}_z^\top R^\top(\mathbf{q}) \tag{74}$$

$$= I - R(\mathbf{q}) \mathbf{e}_z \left(R(\mathbf{q}) \mathbf{e}_z \right)^\top \tag{75}$$

$$= I - \zeta \zeta^\top. \tag{76}$$

This last identity is needed in the derivation of bearing vector kinematics. We have

$$T_\zeta^\top (\zeta^c)^\times (\zeta^c)^\times \mathbf{a} = \left(R(\mathbf{q}) \begin{bmatrix} \mathbf{e}_x & \mathbf{e}_y \end{bmatrix} \right)^\top (\zeta^c)^\times (\zeta^c)^\times \mathbf{a} \tag{77}$$

$$= \begin{bmatrix} \mathbf{e}_x^\top \\ \mathbf{e}_y^\top \end{bmatrix} R^\top(\mathbf{q}) (\zeta^c)^\times (\zeta^c)^\times \mathbf{a} \tag{78}$$

$$= \begin{bmatrix} \mathbf{e}_x^\top \\ \mathbf{e}_y^\top \end{bmatrix} \left(R^\top(\mathbf{q}) \zeta^c \right)^\times \left(R^\top(\mathbf{q}) \zeta^c \right)^\times R^\top(\mathbf{q}) \mathbf{a} \tag{79}$$

$$= \begin{bmatrix} \mathbf{e}_x^\top \\ \mathbf{e}_y^\top \end{bmatrix} (\mathbf{e}_z)^\times (\mathbf{e}_z)^\times R^\top(\mathbf{q}) \mathbf{a} \tag{80}$$

$$= \begin{bmatrix} \mathbf{e}_x^\top \\ \mathbf{e}_y^\top \end{bmatrix} \begin{bmatrix} -\mathbf{e}_x & -\mathbf{e}_y & \mathbf{0} \end{bmatrix} R^\top(\mathbf{q}) \mathbf{a} \tag{81}$$

$$= \begin{bmatrix} -\mathbf{e}_x^\top \mathbf{e}_x & -\mathbf{e}_x^\top \mathbf{e}_y & 0 \\ -\mathbf{e}_y^\top \mathbf{e}_x & -\mathbf{e}_y^\top \mathbf{e}_y & 0 \end{bmatrix} R^\top(\mathbf{q}) \mathbf{a} \tag{82}$$

$$= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} R^\top(\mathbf{q}) \mathbf{a} \tag{83}$$

$$= -\begin{bmatrix} \mathbf{e}_x^\top \\ \mathbf{e}_y^\top \end{bmatrix} R^\top(\mathbf{q}) \mathbf{a} \tag{84}$$

$$= -T_\zeta^\top \mathbf{a}, \tag{85}$$

which means that

$$T_\zeta^\top (\zeta^c)^\times (\zeta^c)^\times = -T_\zeta^\top. \tag{86}$$

Landmark kinematics relative to camera

We will derive the feature dynamics relative to the camera. Given a bearing vector ζ^c pointing from the camera to a landmark, the landmark location can be defined in the inertial frame by

$$\mathbf{p}_{il}^i = \mathbf{p}_{ic}^i + R\left(\mathbf{q}_c^i\right) \zeta^c d(\rho). \tag{87}$$

Taking the time derivative of the entire equation yields

$$0 = \mathbf{v}_{c/i}^i + \frac{d}{dt} \left(R\left(\mathbf{q}_c^i\right) \right) \zeta^c d(\rho) + R\left(\mathbf{q}_c^i\right) \frac{d}{dt} (\zeta^c) d(\rho) + R\left(\mathbf{q}_c^i\right) \zeta^c \frac{d}{dt} (d(\rho)) \tag{88}$$

$$= \mathbf{v}_{c/i}^i + R\left(\mathbf{q}_c^i\right) \left(\omega_{c/i}^c \right)^\times \zeta^c d(\rho) + R\left(\mathbf{q}_c^i\right) (\zeta^c)^\times T_\zeta \dot{\zeta}^c d(\rho) + R\left(\mathbf{q}_c^i\right) \zeta^c d'(\rho) \dot{\rho}, \tag{89}$$

and rotating into the camera frame gives

$$0 = \mathbf{v}_{c/i}^c + \left(\boldsymbol{\omega}_{c/i}^c\right)^\times \zeta^c d(\rho) + (\zeta^c)^\times T_\zeta \dot{\zeta}^c d(\rho) + \zeta^c d'(\rho)\dot{\rho} \tag{90}$$

$$= \mathbf{v}_{c/i}^c - (\zeta^c)^\times \boldsymbol{\omega}_{c/i}^c d(\rho) + (\zeta^c)^\times T_\zeta \dot{\zeta}^c d(\rho) + \zeta^c d'(\rho)\dot{\rho}. \tag{91}$$

In order to isolate $\dot{\zeta}^c$, multiply by $d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times$ to get

$$0 = d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times \mathbf{v}_{c/i}^c - d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times (\zeta^c)^\times \boldsymbol{\omega}_{c/i}^c d(\rho) + \tag{92}$$
$$d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times (\zeta^c)^\times T_\zeta \dot{\zeta}^c d(\rho) + d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times \zeta^c d'(\rho)\dot{\rho}$$

$$= d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times \mathbf{v}_{c/i}^c - T_\zeta^\top (\zeta^c)^\times (\zeta^c)^\times \boldsymbol{\omega}_{c/i}^c + T_\zeta^\top (\zeta^c)^\times (\zeta^c)^\times T_\zeta \dot{\zeta}^c, \tag{93}$$

and using the identity $T_\zeta^\top (\zeta^c)^\times (\zeta^c)^\times = -T_\zeta^\top$ followed by $T_\zeta^\top T_\zeta = I$ gives us

$$0 = d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times \mathbf{v}_{c/i}^c + T_\zeta^\top \boldsymbol{\omega}_{c/i}^c - T_\zeta^\top T_\zeta \dot{\zeta}^c \tag{94}$$

$$= d(\rho)^{-1} T_\zeta^\top (\zeta^c)^\times \mathbf{v}_{c/i}^c + T_\zeta^\top \boldsymbol{\omega}_{c/i}^c - \dot{\zeta}^c. \tag{95}$$

Solving for $\dot{\zeta}^c$ yields

$$\dot{\zeta}^c = T_\zeta^\top \left(\boldsymbol{\omega}_{c/i}^c + (\zeta^c)^\times \frac{\mathbf{v}_{c/i}^c}{d(\rho)}\right) \tag{96}$$

$$= T_\zeta^\top \left(\boldsymbol{\omega}_{c/i}^c + \rho (\zeta^c)^\times \mathbf{v}_{c/i}^c\right). \tag{97}$$

To isolate $\dot{\rho}$, multiply (91) by $d'(\rho)^{-1} (\zeta^c)^\top$ to get

$$0 = d'(\rho)^{-1} (\zeta^c)^\top \mathbf{v}_{c/i}^c - d'(\rho)^{-1} (\zeta^c)^\top (\zeta^c)^\times \boldsymbol{\omega}_{c/i}^c d(\rho) + \tag{98}$$
$$d'(\rho)^{-1} (\zeta^c)^\top (\zeta^c)^\times T_\zeta \dot{\zeta}^c d(\rho) + d'(\rho)^{-1} (\zeta^c)^\top \zeta^c d'(\rho)\dot{\rho}$$

$$= d'(\rho)^{-1} (\zeta^c)^\top \mathbf{v}_{c/i}^c + \dot{\rho}, \tag{99}$$

and solving for $\dot{\rho}$ yields;

$$\dot{\rho} = -\frac{(\zeta^c)^\top \mathbf{v}_{c/i}^c}{d'(\rho)} \tag{100}$$

$$= \rho^2 (\zeta^c)^\top \mathbf{v}_{c/i}^c. \tag{101}$$

Camera kinematics

The position of the camera in the inertial frame is given by

$$\mathbf{p}_{ic}^i = \mathbf{p}_{ib}^i + R\left(\mathbf{q}_b^i\right) \mathbf{p}_{bg}^b + R\left(\mathbf{q}_b^i\right) R\left(\mathbf{q}_g^b\right) \mathbf{p}_{gc}^g, \tag{102}$$

and assuming that the camera frame is centered on and aligned with the gimbal frame, we are left with

$$\mathbf{p}_{ic}^i = \mathbf{p}_{ib}^i + R\left(\mathbf{q}_b^i\right) \mathbf{p}_{bc}^b. \tag{103}$$

Differentiating gives

$$\dot{\mathbf{p}}_{ic}^i = \dot{\mathbf{p}}_{ib}^i + \dot{R}\left(\mathbf{q}_b^i\right) \mathbf{p}_{bc}^b + R\left(\mathbf{q}_b^i\right) \dot{\mathbf{p}}_{bc}^b, \tag{104}$$

and assuming a fixed camera, yields

$$\dot{\mathbf{p}}_{ic}^{i} = \dot{\mathbf{p}}_{ib}^{i} + \dot{R}\left(\mathbf{q}_{b}^{i}\right)\mathbf{p}_{bc}^{b} \tag{105}$$

$$\mathbf{v}_{c/i}^{i} = \mathbf{v}_{b/i}^{i} + R\left(\mathbf{q}_{b}^{i}\right)\left(\omega_{b/i}^{b}\right)^{\times}\mathbf{p}_{bc}^{b} \tag{106}$$

Rotating both sides into the camera frame gives

$$R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{b}^{cb}\right)R\left(\mathbf{q}_{i}^{b}\right)\mathbf{v}_{c/i}^{i} = R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{b}^{cb}\right)R\left(\mathbf{q}_{i}^{b}\right)\mathbf{v}_{b/i}^{i} + R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{b}^{cb}\right)R\left(\mathbf{q}_{i}^{b}\right)R\left(\mathbf{q}_{b}^{i}\right)\left(\omega_{b/i}^{b}\right)^{\times}\mathbf{p}_{bc}^{b} \tag{107}$$

$$\mathbf{v}_{c/i}^{c} = R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{b}^{cb}\right)\mathbf{v}_{b/i}^{b} + R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{b}^{cb}\right)\left(\omega_{b/i}^{b}\right)^{\times}\mathbf{p}_{bc}^{b} \tag{108}$$

$$\mathbf{v}_{c/i}^{c} = R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{b}^{cb}\right)\left(\mathbf{v}_{b/i}^{b} + \left(\omega_{b/i}^{b}\right)^{\times}\mathbf{p}_{bc}^{b}\right) \tag{109}$$

$$\mathbf{v}_{c/i}^{c} = R\left(\mathbf{q}_{b}^{c}\right)\left(\mathbf{v}_{b/i}^{b} + \left(\omega_{b/i}^{b}\right)^{\times}\mathbf{p}_{bc}^{b}\right). \tag{110}$$

The angular velocity of the camera is the combination of body rotation and gimbal rotation given by

$$\omega_{c/i}^{c} = R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{g}^{cb}\right)R\left(\mathbf{q}_{g/b}\right)\omega_{b/i}^{b} + R\left(\mathbf{q}_{cb}^{c}\right)R\left(\mathbf{q}_{g}^{cb}\right)\omega_{g/b}^{g} \tag{111}$$

$$= R\left(\mathbf{q}_{b}^{c}\right)\omega_{b/i}^{b} + R\left(\mathbf{q}_{g}^{c}\right)\omega_{g/b}^{g}, \tag{112}$$

and since we are assuming a fixed camera, this becomes

$$\omega_{c/i}^{c} = R\left(\mathbf{q}_{b}^{c}\right)\omega_{b/i}^{b}. \tag{113}$$

Jacobians of state kinematics

The state and input noise for a single landmark estimate is given by

$$\mathbf{x} = \begin{bmatrix}\mathbf{q}^{\top} & \rho\end{bmatrix}^{\top} \tag{114}$$

$$\mathbf{u} = \begin{bmatrix}\eta_{\omega}^{\top} & \eta_{a}^{\top}\end{bmatrix}^{\top}. \tag{115}$$

The true body-relative state kinematics are given by

$$\dot{\mathbf{q}} = T_{\zeta}^{\top}\left(\omega_{c/i}^{c} + \rho\left(\zeta^{c}\right)^{\times}\mathbf{v}_{c/i}^{c}\right) \tag{116}$$

$$\dot{\rho} = \rho^{2}\left(\zeta^{c}\right)^{\top}\mathbf{v}_{c/i}^{c}, \tag{117}$$

where $\omega_{b/i}^{b}$ and $\mathbf{v}_{b/i}^{b}$ are inputs from the tracking vehicle's state estimator and

$$\mathbf{v}_{c/i}^{c} = R\left(\mathbf{q}_{b}^{c}\right)\left(\mathbf{v}_{b/i}^{b} + \left(\omega_{b/i}^{b}\right)^{\times}\mathbf{p}_{bc}^{b}\right) \tag{118}$$

$$\omega_{c/i}^{c} = R\left(\mathbf{q}_{b}^{c}\right)\omega_{b/i}^{b}. \tag{119}$$

The covariance of the state kinematics w.r.t. the state is then given by

$$F = \frac{\partial\dot{\mathbf{x}}}{\partial\mathbf{x}} = \begin{bmatrix}\frac{\partial\dot{\mathbf{q}}}{\partial\mathbf{q}} & \frac{\partial\dot{\mathbf{q}}}{\partial\rho} \\ \frac{\partial\dot{\rho}}{\partial\mathbf{q}} & \frac{\partial\dot{\rho}}{\partial\rho}\end{bmatrix}, \tag{120}$$

where

$$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} = -T_\zeta^\top \left(\left(\boldsymbol{\omega}_{c/i}^c + \rho \left(\zeta^c\right)^\times \mathbf{v}_{c/i}^c \right)^\times + \rho \left(\mathbf{v}_{c/i}^c \right)^\times \left(\zeta^c\right)^\times \right) T_\zeta \tag{121}$$

$$\frac{\partial \dot{\mathbf{q}}}{\partial \rho} = T_\zeta^\top \left(\zeta^c\right)^\times \mathbf{v}_{c/i}^c \tag{122}$$

$$\frac{\partial \dot{\rho}}{\partial \mathbf{q}} = \rho^2 \left(\mathbf{v}_{c/i}^c \right)^\top \left(\zeta^c\right)^\times T_\zeta \tag{123}$$

$$\frac{\partial \dot{\rho}}{\partial \rho} = 2\rho \left(\zeta^c\right)^\top \mathbf{v}_{c/i}^c. \tag{124}$$

The covariance of the state dynamics w.r.t. the input noise is given by

$$G = \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}} = \begin{bmatrix} \dfrac{\partial \dot{\mathbf{q}}}{\partial \eta_\omega} & \mathbf{0} \\ \dfrac{\partial \dot{\rho}}{\partial \eta_\omega} & \mathbf{0} \end{bmatrix}, \tag{125}$$

where

$$\frac{\partial \dot{\mathbf{q}}}{\partial \eta_\omega} = T_\zeta^\top \left(-R\left(\mathbf{q}_b^c\right) + \rho \left(\zeta^c\right)^\times R\left(\mathbf{q}_b^c\right) \left(\mathbf{p}_{bc}^b \right)^\times \right) \tag{126}$$

$$\frac{\partial \dot{\rho}}{\partial \eta_\omega} = \rho^2 \left(\zeta^c\right)^\top R\left(\mathbf{q}_b^c\right) \left(\mathbf{p}_{bc}^b \right)^\times. \tag{127}$$

ACKNOWLEDGMENTS

This research was supported through the U.S. Department of Defense SMART Scholarship program and 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.

REFERENCES

[1] Douglas W Murphy and James Cycon. Applications for mini vtol uav for law enforcement. In *Sensors, C3I, Information, and Training Technologies for Law Enforcement*, volume 3577, pages 35–44. International Society for Optics and Photonics, 1999.

[2] Inseok Hwang, Hamsa Balakrishnan, Kaushik Roy, and Claire Tomlin. Multiple-target tracking and identity management in clutter, with application to aircraft tracking. In *American Control Conference, 2004. Proceedings of the 2004*, volume 4, pages 3422–3428. IEEE, 2004.

[3] Ponciano Jorge Escamilla-Ambrosio and N Lieven. A multiple-sensor multiple-target tracking approach for the autotaxi system. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 601–606. IEEE, 2004.

[4] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007.

[5] Brandon A Jones, Daniel S Bryant, Ba-Tuong Vo, and Ba-Ngu Vo. Challenges of multi-target tracking for space situational awareness. In *Information Fusion (FUSION), 2015 18th International Conference on*, pages 1278–1285. IEEE, 2015.

[6] Donald Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979.

[7] Thomas E Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Multi-target tracking using joint probabilistic data association. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, pages 807–812. IEEE, 1980.

[8] B-N Vo and W-K Ma. The gaussian mixture probability hypothesis density filter. *IEEE Transactions on signal processing*, 54(11):4091–4104, 2006.

[9] Peter C Niedfeldt. *Recursive-RANSAC: A novel algorithm for tracking multiple targets in clutter*. PhD thesis, 2014. Available at `http://scholarsarchive.byu.edu/etd/4195/`.

[10] Peter C Niedfeldt, Kyle Ingersoll, and Randal W Beard. Comparison and analysis of recursive-ransac for multiple target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 53(1):461–476, 2017.

[11] Peter C Niedfeldt and Randal W Beard. Convergence and complexity analysis of recursive-ransac: a new multiple target tracking algorithm. *IEEE Transactions on Automatic Control*, 61(2):456–461, 2016.

[12] Kyle Ingersoll, Peter C Niedfeldt, and Randal W Beard. Multiple target tracking and stationary object detection in video with recursive-ransac and tracker-sensor feedback. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 1320–1329. IEEE, 2015.

[13] Peter C Niedfeldt and Randal W Beard. Multiple target tracking using recursive ransac. In *American Control Conference (ACC), 2014*, pages 3393–3398. IEEE, 2014.

[14] Joshua Y Sakamaki, Randal W Beard, and Michael D Rice. Cooperative estimation for a vision-based target tracking system. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pages 878–885. IEEE, 2016.

[15] Randal W Beard and Timothy W McLain. *Small unmanned aircraft: Theory and practice*. Princeton university press, 2012.

[16] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 298–304. IEEE, 2015.

[17] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

[18] Michael Andre Bloesch. *State Estimation for Legged Robots–Kinematics, Inertial Sensing, and Computer Vision*. PhD thesis, 2017. Available at `https://www.research-collection.ethz.ch/handle/20.500.11850/129873`.