2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Sense and Avoid Based on Visual Pose Estimation for Small UAS

Changkoo Kang,[1] Jason Davis,[2] Craig A. Woolsey[3] and Seongim Choi[4]

Abstract— **Small unmanned aircraft systems (UAS) must be able to detect and avoid conflicting traffic, an especially challenging task when the threat is another small UAS. Collision avoidance requires trajectory prediction and the performance of a collision avoidance system can be improved by extending the prediction horizon. We describe an algorithm that predicts the trajectory of a small, fixed-wing UAS using an estimate of its orientation. First, a computer vision algorithm locates specific feature points of the threat aircraft in an image. Next, the POSIT algorithm uses these feature points to estimate the pose (position and attitude) of the threat. A sequence of pose estimates is then used to predict the trajectory of the threat aircraft in order to avoid a collision. To assess the algorithm's performance, the predictions are compared with predictions based solely on position estimates for a variety of encounter scenarios. Simulation and experimental results indicate that trajectory prediction using orientation estimates provides quicker response to a change in the threat aircraft trajectory and better prediction and avoidance performance.**

I. Introduction

As the private and commercial use of small unmanned aircraft systems (UAS) continues to expand, low altitude air traffic will become more congested raising the risk of mid-air collisions that may result in injuries to people or damage to property below. For many years, commercial manned aircraft have used the Traffic Collision Avoidance System (TCAS) to help ensure that aircraft do not collide in flight [1]. TCAS and similar collision avoidance systems are useful, however, only when every aircraft in the airspace uses the technology. Small UAS typically operate at low altitude where collision threats include general aviation aircraft and other UAS, which may not carry their own collision avoidance equipment [2]. In these scenarios, it is urgent that the small unmanned aircraft be able to sense and avoid these threats.

Sense and Avoid (SAA) technologies have been developed to mitigate the risk of collision between UAS and other aircraft [3], [4]. Researchers have explored other active communication-based methods besides TCAS, such as Automatic Dependent Surveillance-Broadcast (ADS-B) [3], but any method that depends on standardized communication equipment is inherently susceptible to non-compliant aircraft. Active sensing, such as radar [5] and lidar [6], has been investigated for collision avoidance, but these sensors are typically expensive, heavy, and power hungry, at least at the scale of a small UAS. Cameras provide a low-cost, low-power alternative and the technology is advancing at a rapid pace thanks to intense competition within the consumer electronics industry. While the utility of cameras is limited to visual flight rules (VFR) conditions, these conditions are typical for current small UAS operations. Here, we investigate the ability of a single, monocular camera to provide actionable SAA data for a fixed-wing threat aircraft. To bound the scope of the problem, we assume the image quality is sufficient to enable classification and, moreover, that this classification task has been accomplished so that the threat aircraft is known. Given this knowledge of the threat, the remaining challenge is to predict its motion.

For a camera with sufficient resolution, an unobstructed image can provide detail that other sensors, such as radar or lidar, cannot provide. Earlier work using cameras [7], [8], [9] sought to determine direction (azimuth and elevation) to a threat, but not orientation. As the size, weight, and power and the cost (SWaP-C) of cameras has continued to drop, while resolution and image quality has continued to improve, and as computer vision methods have continued to develop, it has become possible to extract more information (e.g., aircraft attitude) from an image than was possible in the past.

If a threat aircraft flies along a straight path, it is not hard to predict its future trajectory using only position data. If the threat begins to turn, however, then trajectory predictions based solely on position will accrue error; if the error is sufficiently large, it could compromise the prediction algorithm's ability to inform an avoidance decision. Here, we show that the additional information obtained by estimating the threat aircraft orientation enables a more accurate prediction of the threat aircraft trajectory. Specifically, we present an algorithm that estimates the threat aircraft's roll angle and uses this information, together with a velocity estimate, to estimate the turn rate. The turn rate is then used to predict the trajectory over a fixed time horizon. The prediction method compares favorably with a more conventional method that uses only position data.

Section II describes the computer vision methods, including feature point detection and pose estimation. Section III describes the aircraft motion model and two approaches to prediction: the proposed approach and a more conventional approach for comparison. An avoidance algorithm is presented in Section IV and Section V describes the results from simulations and flight tests. Finally, Section VI presents conclusions and summarizes ongoing work.

[1]Graduate Research Assistant, Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA changkk@vt.edu
[2]Undergraduate Research Assistant, Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA jdavis7@vt.edu
[3]Professor, Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA cwoolsey@vt.edu
[4]Assistant Professor, Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA schoi1@vt.edu

978-1-5386-2682-5/17/$31.00 ©2017 IEEE

II. Computer Vision Technology

In order to determine the pose of a fixed-wing aircraft observed by a camera, we first use a computer vision algorithm to detect five distinguished feature points: the nose, the two wing tips, and the two tips of the horizontal stabilizer. These points are assumed to be visible in the image; the effect of occluded feature points is a topic of ongoing study. It is also assumed that the geometry of the observed aircraft is known, having been determined, for example, using a pattern recognition algorithm. Finally, the pose estimation algorithm POSIT is used to determine the position and attitude of the observed aircraft.

A. Aircraft feature point detection

Within a given image containing a threat aircraft, the image is cropped to the region of interest, converted to grayscale, and then processed using the Canny Edge Detection algorithm [10]. A modified, bi-directional derivative of the grayscale point values is computed and all pixels below a threshold value are removed. This process leaves a binary image in which points with a value of 1 represent high contrast gradients (i.e., edges).

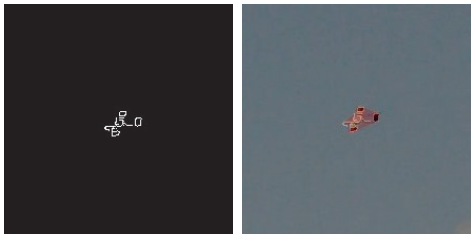

Fig. 1: Aircraft feature point detection

Edges detected using the preceding method are assumed to be either the exterior edges of the aircraft image or aircraft contours contained within the polygon defined by these exterior edges. Next, we find the largest pentagon containing the identified edges [11], [12] and associate the vertices of this pentagon with the five feature points required by the POSIT algorithm. Because the geometry of the threat aircraft is assumed to be known, the pixel distance between these feature points can be used to estimate the distance from the camera to the threat aircraft.

B. Pose estimation

Pose estimation is accomplished using the POSIT algorithm [13], which determines the rotation matrix relating a camera-fixed reference frame to a reference frame fixed in the threat aircraft. We call the threat aircraft frame the "body" frame and let R_{BC} represent the rotation matrix that maps free vectors from the camera frame to the body frame. Similarly, we define a rotation matrix R_{CI} that maps free vectors from the inertial reference frame to the camera-fixed reference frame; we assume that R_{CI} is known, based on sensor data available to the host aircraft. Finally, the orientation of the threat aircraft with respect to inertial space is represented by the rotation matrix R_{BI}, which maps free vectors from the inertial reference frame to the body frame.

$$R_{BI} = R_{BC}R_{CI} \tag{1}$$

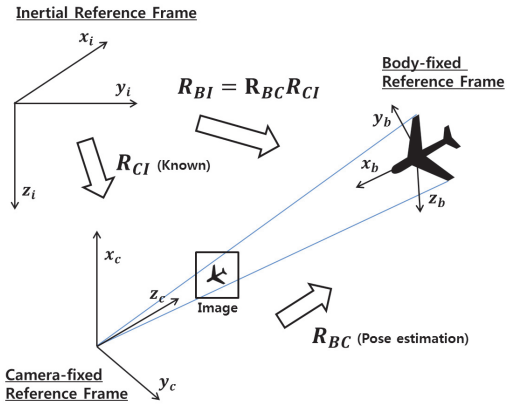

Fig. 2: Pose estimation geometry

III. Trajectory Prediction Algorithm

A. Aircraft model

Having obtained the position and orientation of the threat aircraft, one may estimate the turn rate using the airspeed and roll angle. In this effort, constant-altitude flight is assumed and a two-dimensional coordinated turn (CT) model [14] is used. Although the CT model assumes constant speed and turn rate, these values are updated using vision-based estimates – a "quasi-steady CT" model.

Treating the threat as a particle moving in the horizontal plane, we define the position (x and y), heading (ψ), and speed (v) as well as the wind disturbance components (w_x and w_y). Assuming that the aircraft speed and turn rate $\omega = \dot{\psi}$ remain constant, and defining the state vector $X = [x, \dot{x}, y, \dot{y}]^T$, we obtain the following motion model:

$$\dot{X} = A(\omega)X + W(\omega) \tag{2}$$

where

$$A(\omega) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & 0 & 1 \\ 0 & \omega & 0 & 0 \end{bmatrix} \quad \text{and} \quad W(\omega) = \begin{bmatrix} 0 \\ \omega w_y \\ 0 \\ -\omega w_x \end{bmatrix}$$

The turn rate ω of the threat aircraft varies, but we assume this variation is sufficiently slow that the model (2) remains valid even with varying ω. We estimate the turn rate using two approaches. In the first approach, ω is inferred from the recent position history. In the second, the airspeed and roll angle are estimated in order to compute the turn rate from the CT model. Details are given in Sections III-C and III-B.

B. Position only (PO) approach

Assuming constant-altitude and constant-speed motion, the turn rate at the k^{th} time-step can be inferred from the speed

V and centripetal acceleration $\vec{a_k}$ as follows:

$$\omega_k = \frac{\|\vec{a_k}\|}{V} \tag{3}$$

The resulting turn rate can then be used to predict the trajectory over some time horizon using (2). In implementation, the speed and centripetal acceleration may be estimated numerically from the position history.

C. Position-plus-roll-angle (PPR) approach

Alternatively, the turn rate can be computed using the roll angle of the threat aircraft, as obtained from pose estimation. For a coordinated turn, there is a simple relationship between the turn rate ω and the roll angle ϕ, assuming constant-altitude, constant-speed flight [15]. The "load factor" at the k^{th} time-step is

$$n_k = \frac{1}{\cos\phi_k} \tag{4}$$

and the turn rate is

$$\omega_k = \frac{g\sqrt{n_k^2 - 1}}{V} \tag{5}$$

where g is the specific force of gravity.

IV. Avoidance Algorithm

Assuming the host aircraft flies at constant speed and altitude, we adopt the CT model with turn rate as a control input. To determine avoidance paths, 61 potential host aircraft paths (corresponding to bank angles between $-30°$ and $30°$) are computed and compared with three potential threat trajectories – a nominal prediction and two bounding perturbations that allow for prediction error, to be discussed shortly; see Figure 3.

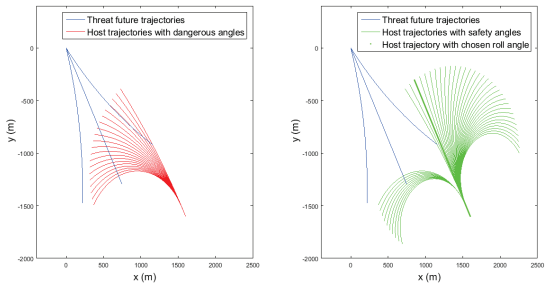

Fig. 3: Potential collision paths (left) and safe paths (right)

Each host and threat aircraft trajectory prediction is a finite time history (10 seconds, in this work). Comparing candidate host trajectories with possible threat trajectories, one may compute a "distance at closest point of approach" denoted d_{CPA}. If d_{CPA} for a given candidate trajectory is less than a threshold value called the safety radius R_d (500 ft, in this case), then the corresponding turn rate (or roll angle) input is considered dangerous. For avoidance, the smallest safe roll angle input is adopted, to minimize the control effort required for the avoidance maneuver.

Once the host aircraft has passed the point of closest approach to the threat aircraft along its avoidance path, the algorithm switches to the trajectory recovery phase. In this phase, the host aircraft follows a line-of-sight guidance strategy to recover its original path.

V. Results

Pose estimation and trajectory prediction were implemented using simulated data as well as experimental data obtained during flight tests. Experimental data were obtained during a Summer 2016 flight test campaign that was aimed at creating a database of visual and radar encounter data for several small UAS [16]. For this paper, the threat aircraft is Virginia Tech's eSPAARO. A PixHawk controller and a Piksi RTK GPS unit mounted on the eSPAARO logged position. In the preliminary tests described here, the Nikon D3200 digital camera used to image the eSPAARO was mounted on a tri-pod, together with a second PixHawk unit, affixed to the camera; a planned flight campaign is aimed at collecting high quality air-to-air imagery.

A. Pose estimation results

The pose estimation experiment was conducted using images obtained from the ground camera system, as described in Section II-B. The results were compared with orientation data from the eSPAARO avionics, expressed using conventional Euler angles. Table I shows the average error magnitude between the orientation estimates from the pose estimation algorithm and direct measurements obtained from 16 encounters during flight tests. The pose estimation algorithm provides reasonably accurate estimates of the threat aircraft attitude. One source of error is the synchronization error between the camera imagery and the camera-mounted PixHawk. Also, the fact that the camera was mounted on the ground resulted in an image set that was less rich than would be obtained using air-to-air imagery. A flight campaign planned for Fall 2017 will obtain a richer image data set.

Fig. 4: Pose estimation

TABLE I: Average error of pose estimation experiments

Value	Roll	Pitch	Yaw
Average Error Magnitude (°)	3.4	2.7	11.6

B. Simulation setup

To assess the proposed prediction algorithm more thoroughly, two simulations were constructed: one for the PO approach and another for PPR approach. The turn rate estimate is a critical parameter for both approaches. In the former case, it was determined solely from the position

history. In the latter case, the estimated roll angle was used as well. Simulations were performed with and without the addition of zero-mean, Gaussian white noise to the position and roll angle. Since the threat aircraft is assumed to be known, there is no additional loss of generality in assuming a known turn rate limit corresponding, for example, to a structural load factor limit at the given speed. Equivalently, we assume a roll angle magnitude limit of $45°$. We also assume a roll *rate* magnitude limit of $10°/\text{s}$.

For the PO approach, the acceleration and velocity are estimated at each time-step using the previous 5 consecutive positions. The turn rate is computed from these values using (6) at each time-step and substituting into (2). For the PPR approach, the turn rate used in (2) is estimated from (5).

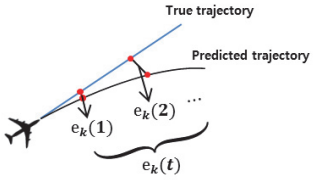

Fig. 5: Instantaneous distance error $e_k(t)$

At each time-step, for each approach, the instantaneous distance error between the predicted trajectory and the true trajectory is computed over a 10 second horizon. The value of this distance error at the k^{th} time-step is denoted $e_k(t)$; see Figure 5. If $e_k(t)$ grows quickly in its argument t, the function predicts a rapid separation between the predicted trajectory and the true trajectory beginning at time-step k.

To approximate $e_k(t)$, various functional forms were evaluated in terms of accuracy and complexity. Figure 6 indicates that a 2^{nd} order polynomial regression provides a good approximation:

$$e_k(t) = \kappa t^2 \tag{6}$$

The coefficient κ then serves as a measure of the growth of e_k. The constant κ is computed for both the PO and PPR approach at each time-step to compare the prediction performance of two approaches and the results are described in the next section.

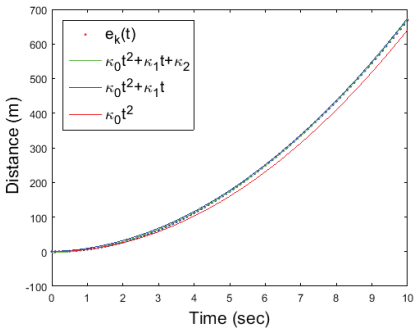

Fig. 6: Representative distance error $e_k(t)$ with 2^{nd} order polynomial regressions.

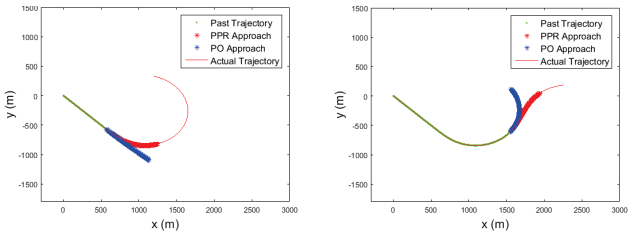

Fig. 7: Simulation examples

C. Prediction simulation results

Figure 8 compares the PO and PPR prediction methods for two simulation cases. Note that κ increases just before time $t = 10$ seconds, when the aircraft begins to turn. When the threat aircraft begins rolling at 10 seconds, both prediction algorithms react to the changing turn rate and correct the prediction so that κ decreases. We call the time taken to correct the predicted trajectory the "prediction correction time." The prediction correction time in Table III indicates that κ in the PPR approach begins to recover, on average, 2.4 seconds faster than the PO approach. For small UAS flying in close proximity, the additional lead time provided by the PPR approach could be critical in making an avoidance decision. For the simulations that include noise, this recovery time is more difficult to define and compute, but a moving average, as shown in Figure 9, suggests the PPR approach again recovers to the true trajectory faster than the PO case. Table III shows the average value of κ for 576 simulations involving various maneuvers in various wind conditions. Note that the average magnitude of κ is lower for the PPR approach than for the PO approach, particularly in the case of noise where the PO approach suffers significantly, as shown in Table II. This result is consistent with the analysis of Section V-D, which suggests that the PO approach is more sensitive to noise than the PPR approach.

TABLE II: Simulation parameters

Parameter	Values
Host aircraft speed (m/s)	100
Threat aircraft speed (m/s)	100
Turn rate change rate (deg/s)	10
Prediction range time (s)	10
Total simulation time (s)	30
Data frequency (Hz)	10
Position noise variance (m^2)	0.5
Roll angle noise variance (deg^2)	3
Wind speed (m/s)	2 - 4
Wind direction (deg)	0 - 360
Host roll angle control range (deg)	[−30 - 30]
Safe radius (m)	152.4

TABLE III: Comparison of prediction correction time and κ

Parameter	PPR	PO
Prediction correction time (s)	4.7	7.1
κ without Noise	1.5	1.6
κ with Noise	1.7	3.3

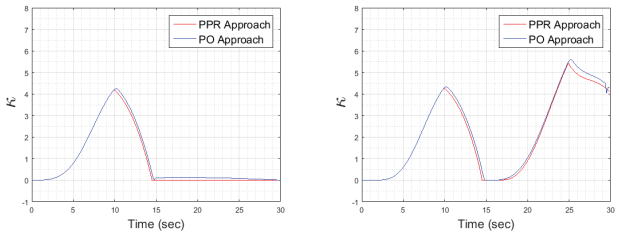

Fig. 8: κ-time graphs without noise

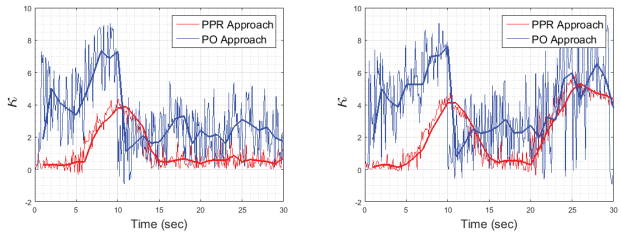

Fig. 9: κ-time graphs with noise

D. Sensitivity to error in variables

In the previous section, the turn rate of the threat aircraft is computed using two different approaches. The resulting turn rate estimates are affected by error in the position and roll angle data. Here we consider sensitivity to these data. For the PO approach, the turn rate is computed using (6). The error in this turn rate estimate due to velocity and acceleration errors (which in turn are due to position data errors) is

$$\begin{aligned}\delta\omega_{PO} &= \frac{\partial}{\partial a_x}\left[\frac{\sqrt{a_x^2+a_y^2}}{\sqrt{v_x^2+v_y^2}}\right]\delta a_x + \frac{\partial}{\partial a_y}\left[\frac{\sqrt{a_x^2+a_y^2}}{\sqrt{v_x^2+v_y^2}}\right]\delta a_y \\ &+ \frac{\partial}{\partial v_x}\left[\frac{\sqrt{a_x^2+a_y^2}}{\sqrt{v_x^2+v_y^2}}\right]\delta v_x + \frac{\partial}{\partial v_y}\left[\frac{\sqrt{a_x^2+a_y^2}}{\sqrt{v_x^2+v_y^2}}\right]\delta v_y \\ &= \frac{1}{\sqrt{a_x^2+a_y^2}\sqrt{v_x^2+v_y^2}}(a_x\delta a_x + a_y\delta a_y) \\ &- \frac{\sqrt{a_x^2+a_y^2}}{(v_x^2+v_y^2)\sqrt{v_x^2+v_y^2}}(v_x\delta v_x + v_y\delta v_y)\end{aligned} \tag{7}$$

For the PPR approach, with the turn rate estimated as in (5), the error in the turn rate estimate due to roll angle and velocity error is

$$\begin{aligned}\delta\omega_{PPR} &= \frac{\partial}{\partial\phi}\left[\frac{g\sqrt{1/\cos^2\phi-1}}{\sqrt{v_x^2+v_y^2}}\right]\delta\phi \\ &+ \frac{\partial}{\partial v_x}\left[\frac{g\sqrt{1/\cos^2\phi-1}}{\sqrt{v_x^2+v_y^2}}\right]\delta v_x \\ &+ \frac{\partial}{\partial v_y}\left[\frac{g\sqrt{1/\cos^2\phi-1}}{\sqrt{v_x^2+v_y^2}}\right]\delta v_y \\ &= \frac{g}{2V}\frac{\tan\phi}{\cos^2\phi\sqrt{1/\cos^2\phi-1}}\delta\phi \\ &- \frac{g\sqrt{1/\cos^2\phi-1}}{(v_x^2+v_y^2)\sqrt{v_x^2+v_y^2}}(v_x\delta v_x + v_y\delta v_y)\end{aligned} \tag{8}$$

Note that $\delta\omega_{PO}$ depends on the acceleration data error, as well as the velocity data error, as computed from the position history. On the other hand, $\delta\omega_{PPR}$ depends only on the velocity data error (along with roll angle error). A formal error comparison was not performed, but simulations indicate the PO approach is much more sensitive to typical data errors than the PPR approach; see Table IV.

E. Trajectory prediction experiments

The prediction algorithm was also applied to the trajectory data obtained in flight tests [16]. In all, sixteen segments

TABLE IV: Average turn rate error magnitude

Parameter	PPR	PO
Average Turn Rate Error Magnitude (deg/s)	0.9	4.8

of a sample trajectory which was obtained from flight tests were considered (Figure 10) and the algorithms described above were applied to each segment. Figure 10 shows two of sixteen segments and the prediction algorithm is applied to each path. The thick red curve represents the predicted trajectory based on the PPR approach and the thick blue curve is a predicted trajectory based on the PO approach. A thin blue line represents the actual path. In this figure, the PPR prediction is closer to the actual data than the PO prediction. The κ-time graphs in Figure 11 also indicate the PPR approach provides better prediction performance since κ is generally smaller for the PPR approach. Table V shows the

TABLE V: Average magnitude of κ for flight tests

Parameter	PPR	PO
Average κ	7.3	8.7

average magnitude of κ from the flight tests we implemented. The value for the two approaches is not as low as it was in simulation. It is important to note that the eSPAARO did not fly with a truly constant speed nor at a truly constant altitude, as assumed in the CT model. In any case, the PPR approach results in lower values for κ than the PO approach, suggesting that trajectory prediction informed by orientation estimates can be more accurate than prediction based solely on position measurements.

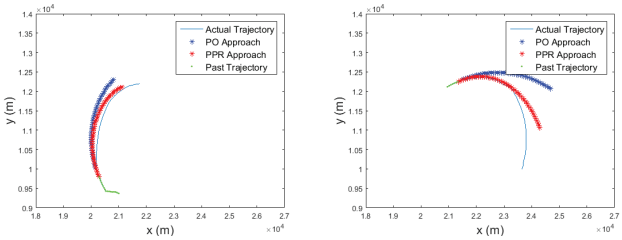

Fig. 10: Extracted experiment trajectories

F. Avoidance simulation results

To check the avoidance performance based on the PPR approach and the PO approach, a host aircraft running the

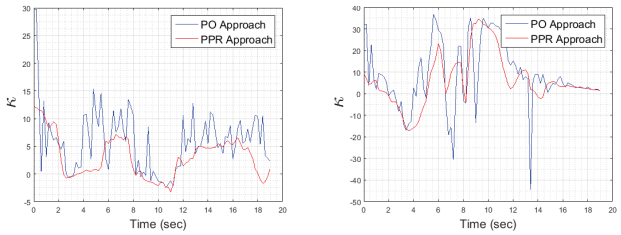

Fig. 11: κ-time graphs of extracted trajectories

avoidance algorithm was added to the prediction simulation. As described in the previous section, both the threat aircraft and the host aircraft use the same motion model. Wind is included in the simulation. The speed of both aircraft is 100 m/s, the total simulation time is 60 seconds, and the trajectory prediction time is 10 seconds.

Figures 12 depicts the paths of a host and threat aircraft, with the host aircraft successfully sensing and avoiding the threat using the PO and PPR approaches, respectively, and then recovering its original trajectory using the avoidance algorithm.

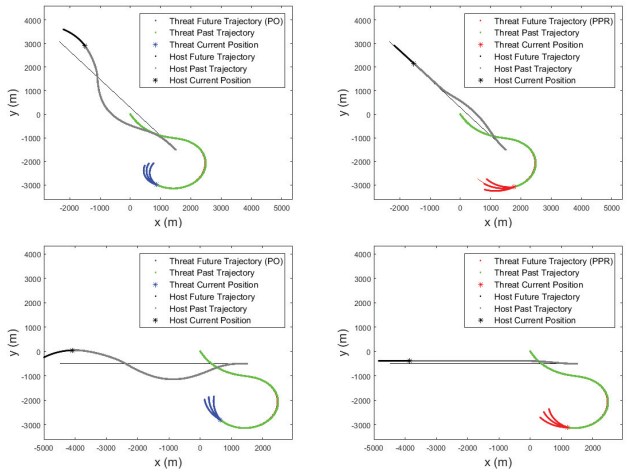

Fig. 12: Avoidance simulations for the PO approach (left) and PPR approach (right)

The representative examples shown in Figure 12 indicates that avoidance based on the PPR approach is safer than that based on the PO approach. As discussed earlier, PO trajectory predictions for the threat aircraft are less accurate and oscillate more than PPR predictions, making avoidance more challenging using the PO approach.

VI. Conclusion

An algorithm was presented for predicting the trajectory of a fixed-wing aircraft using vision-based estimates of the aircraft orientation. The aircraft orientation is estimated by identifying feature points and applying the POSIT algorithm. The prediction method incorporates the orientation data into a coordinated turn model for aircraft motion to predict the trajectory. Prediction results compared favorably with results based solely on the position history, both in accuracy and speed of response to threat aircraft maneuvers, and the PPR approach appears less sensitive to noisy data. Moreover, in simulations of a simple avoidance algorithm, predictions based on the PPR approach yielded better avoidance performance than those based on the PO approach.

While results from analysis, simulation, and experiments are promising, the vision-based prediction algorithm is designed for a specific use case under restrictive assumptions – constant speed and altitude of the threat aircraft. Ongoing work aims to refine and improve the algorithm by relaxing these assumptions and also by expanding the experimental validation data set to include high quality air-to-air encounter imagery. Additional challenges include extending the approach to multirotor UAS and to unconventional configurations.

Acknowledgment

The authors gratefully acknowledge support from the Center for UAS under NSF Grant Nos. IIP-1539975 and CNS-1650465.

References

[1] Olson, W. A., & Olszta, J. E. (2010). TCAS operational performance assessment in the US national airspace. *Digital Avionics Systems Conference (DASC), 2010 IEEE/AIAA 29th.*

[2] Department of Transportation, Federal Aviation Administration. (2013). *Pilot's Handbook of Aeronautical Knowledge*.

[3] Zhao, C., Gu, J., Hu, J., Lyu, Y., & Wang, D. (2016). Research on cooperative sense and avoid approaches based on ADS-B for unmanned aerial vehicle. *Guidance, Navigation and Control Conference (CGNCC), 2016 IEEE Chinese*.

[4] Yu, X., & Zhang, Y. M. (2015). Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Progress in Aerospace Sciences, 74*, 152-166.

[5] Sahawneh, L. R., Spencer, J., Beard, R. W., & Warnick, K. (2016). Minimum required sensing range for UAS sense and avoid systems. *AIAA Infotech@ Aerospace, 1982*.

[6] Ramasamy, S., Sabatini, R., Gardi, A., & Liu, J. (2016). LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid. *Aerospace Science and Technology, 55*, 344-358.

[7] Sapkota, K. R., Roelofsen, S., Rozantsev, A., Lepetit, V., Gillet, D., Fua, P., & Martinoli, A. (2016). Vision-based unmanned aerial vehicle detection and tracking for sense and avoid systems. *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*.

[8] Mcfadyen, A., & Mejias, L. (2016). A survey of autonomous vision-based See and Avoid for Unmanned Aircraft Systems. *Progress in Aerospace Sciences, 80*, 1-17.

[9] Lyu, Y., Pan, Q., Zhao, C., & Hu, J. (2017). Autonomous stereo vision based collision avoid system for small UAV. *AIAA Information Systems-AIAA Infotech@ Aerospace*, 1150.

[10] Zhao, X. M., Wang, W. X., & Wang, L. P. (2011). Parameter optimal determination for canny edge detection. *Imaging Science Journal, 59*(6), 332-341

[11] Persson, A. H., Bondesson, L., & Börlin, N. (2006). Estimation of Polygons and Areas. *Scandinavian Journal Of Statistics, 33*(3), 541-559.

[12] Ratschek, H., Rokne, J., & Yap, C. (2000). Exact and Optimal Convex Hulls in 2D. *International Journal Of Computational Geometry and Applications, 10*(2), 109.

[13] DeMenthon, D. F., & Davis, L. S. (1992). Model-based object pose in 25 lines of code. *European Conference on Computer Vision*.

[14] Li, X. R., & Jilkov, V. P. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems, 39*(4), 1333-1364.

[15] John, A. D., & Anderson, J. (1989). *Introduction to Flight, 3rd Edition*. New York, NY:McGraw-Hill, 326-330.

[16] McClelland, H. G., Kang, C., Woolsey, C. A., Roberts, A. K., Buck, D., Cheney, T., & Warnick, K. (2017). Small Aircraft Flight Encounters Database for UAS Sense and Avoid. *AIAA Information Systems-AIAA Infotech@ Aerospace*, 1152.