An Improved Far-Field Small Unmanned Aerial System Optical Detection Algorithm

Chester Dolph
Aeronautic Systems Engineering Branch
NASA
Hampton, VA USA
chester.v.dolph@nasa.gov

Louis Glaab
Aeronautic Systems Engineering Branch
NASA
Hampton, USA
louis.j.glaab.@nasa.gov

Bonnie Allen
Autonomous Integrated Systems Reseach Branch
NASA
Hampton, USA
danette.allen@nasa.gov

Maria Consiglio
Safety-Critical Avionics Systems Branch
NASA
Hampton, USA
maria.c.consiglio@nasa.gov

Khan Iftekharuddin
Old Dominion University
Batten College of Engineering and Technology
Norfolk, USA
kiftekha@odu.edu

***Abstract*— Onboard far-field aircraft detection is needed for safe non-cooperative traffic mitigation for autonomous small Unmanned Aerial System (sUAS) operations. This work presents an aircraft detect and track pipeline that fuses image differencing and morphological filtering detections for inputs to a Kalman-based object tracking pipeline. The pipeline is evaluated using two types of flight encounters: 1) motorcopter sUAS vs. fixed-wing sUAS 2) motorcopter sUAS vs general aviation plane.**

Keywords—sense and avoid, detect and track small targets, Unmanned Aerial Vehicle collision avoidance

I. Introduction

The rapid expansion of sUAS commercial applications such as package delivery, aerial inspection, and emergency response has created a need for safe, on-board traffic avoidance systems [1]. In 2022, the number of private (or "model") sUAS in the United States may reach as many as 3.17 million and the number of commercial sUAS 0.717 million [2], which corresponds to 2.9x growth for hobby sUAS over 5 years and 6.5x growth for commercial sUAS. The sUAS traffic management is a challenging problem because of low altitudes, shared airspace for general aviation and sUAS, cooperative and non-cooperative sUAS (RF communication links to de-conflict traffic), and the desire for high density of sUAS operating in a small airspace for urban environment for applications such as package delivery. To this end, the FAA and NASA are working together, along with private industry, to develop a Concept of Operations (ConOps) for the UAS Traffic Management (UTM), which focuses on traffic management below altitudes of 400 ft.

sUAS traffic management techniques include wireless communication links, radar based autonomous Sense and Avoid (SAA), infrared based SAA, and optical based SAA. Each of these techniques has their strengths and weaknesses. For instance, on-board wireless communication may suffer from dropout in cluttered environments from urban canyons and ground-based wireless communication systems (e.g. cellular networks) may have too much latency for safety critical applications such as sUAS missions. On-board radar sensors tend to be heavy relative to sUAS payloads and ground-based radar systems would require substantial new infrastructure. Infrared and optical sensors require line of sight and produce degraded information in weather conditions such as fog or cloud cover.

This work presents a pipeline for low-altitude far-field sUAS detection and tracking. This detect and track pipeline uses both image-differencing and morphological detection techniques with a Kalman-filtering based tracker. The pipeline is evaluated for two cases: 1) fixed-wing sUAS vs motorcopter camera carrier 2) General Aviation (GA) SR22 vs motorcopter camera carrier. This paper is organized as follows: background section summarizing alternate sUAS detection techniques and prior work completed at NASA, experimental design describing aircraft, hardware, and flight paths, the detect and track pipeline, results, conclusion, and future work.

II. Background

A. Optical-based detection and tracking techniques.

The authors in [3] compared Electro-Optical (EO) based tracking systems to pulsed radar in manned aircraft SAA using 16 head-on encounter scenarios with varying horizontal separation distances, vertical separation distances, and weather conditions. The findings were: 1) radar had decreased sensitivity to changes in weather conditions 2) EO systems had superior range and sensitivity in favorable conditions 3) First detection distances for both radar and EO system varied substantially across the encounter scenarios, which were attributed to the variations in flight encounter approaches and weather conditions, and 4) EO systems detected the intruder before the radar system in 6 of the 16 encounter scenarios.

However, the radar was pre-tuned to the specific manned intruder aircraft and was not generalized for any intruder aircraft (e.g. other manned aircraft or sUAS). The EO detection algorithm used morphological filtering with local-image analysis and mutiframe stabilization. Tracking was achieved through Extended Kalman Filter (EKF). An additional method was presented which fused radar-EO data by a centralized EKF.

The authors in [4] propose that EO sensors are well-suited for SAA on sUAS due to their Size, Weight, and Power (SWAP) properties. Their SAA system was mounted in a ScanEagle (military grade sUAS) and the intruder aircraft was a Cessna 172R. The components of the SAA are: 5 Megapixels (MP) camera, Global Positioning System Inertial Navigation (GPS/INS) unit, single board computer, hard drive for recording data, and a Wi-Fi modem to transmit video and results from real-time onboard SAA. The detection algorithm used GPS/INS based image stabilization to mitigate aircraft vibration, wind gusts, and turbulence. Point features were extracted from the stabilized imagery using a bottom-hat morphological filter and four independent Hidden Markov Models (HMM) were applied for temporal filtering. The detections were then thresholded based on the number of detections of consecutive frames to suppress false positives. They tested their system using 22 head-on collision-course encounters (15 with 5 mm lens, 4 with 8 mm, and 3 with 5 mm). Average detection ranges were 1.48 km and 2.14 km for the 5mm and 8 mm lens respectively, which corresponds to 19.2 and 27.7 seconds until collision based on 150 kt. closing speed. This work used an altitude separation of 500 ft. and the intruder aircraft was always flown above the horizon.

The authors in [5] collected video data of manned aircraft for three clear and four cloudy days for head-on and tail-chase scenarios. Their method achieved no false alarms in 6.63 hours out of 14.14 hours of flight data. Like the authors in [4], the authors used GPS-INS based image-stabilization as part of their image pre-processing pipeline. The authors used two types of temporal processing: Viterni-based filtering and HMM for object detection. These methods both entailed image filtering with temporal analysis and pre-defined thresholds for successful object detection. The closing speed for the head-on collision was 200 knots while the closing speed for the tail-chase was approximately 15 knots.

The authors in [6] performed detection and tracking on manned aircraft in four head-on and six crossing scenarios. Their first detection ranged from 1350 to 2960 meters and their algorithm achieved tracking at distances ranging from 980 to 2588 meters. The authors first determined the location of the horizon in the imagery with the aid of a GPS Inertial Measurement Unit (IMU) sensor. For above the horizon, the authors used morphological close-minus-open filters. For below the horizon, the authors used a differencing pipeline where keypoints are extracted and tracked between frames. The authors found that the morphological feature detector generated many false positives when used below the horizon and that the difference pipeline did not achieve as early of detection when used above the horizon. An Extended Kalman Filter (EKF) was used to assign the detections to tracks.

The authors in [7] applied a deep learning technique on a data set acquired from a delta wing sUAS imaging up to 8 sUAS simultaneously. Low resolution GoPro cameras with fisheye lenses were used and initial inspection of the dataset suggests lower clutter relative to the two NASA test sites later described in this paper. The details regarding the flight patterns, altitudes, intruder sUAS specifications, sUAS speed and separation distances are not stated. Their detection and classification pipeline is as follows: first estimate background movement using a perspective transformation model, compute the difference image perspective transformation for two consecutive images, determine salient points using the Shi-Tomasi corner detector, extract a 40 by 40 patch in the background image for each salient point, and classify patches using a deep convolutional neural network. The intruders are tracked using a Kalman filter. The detection accuracies for their neural network approach are: precision 0.819 ± 0.09, recall 0.798 ± 0.10, and F-Score 0.806 ± 0.08.

B. *Summary of Prior Work: NASA 2015 Dataset*

The NASA 2015 dataset in [8] utilized fisheye lens with 4k action cameras 170 Field of View from aerial sUAS vs. sUAS object detection. The camera carrier sUASs included motorcopter and fixed-wing sUAS. The intruder sUAS included motorcopter sUAS and two types of fixed-wing sUAS. The intruder sUAS was not visible for a large portion of this dataset due to stock action camera sensor and lens limitations for these separation distances. An image differencing approach utilizing FAST features generated limited detection results [8]. Most detections occurred when the intruder sUAS was performing a high profile maneuver above the horizon and near the minimum separation distance. The work in [9] reports a high sUAS detection rate with a very high false positive rate for one of the videos using the NASA 2015 dataset. The dataset is available for download in [10].

C. *Research Gap*

A robust solution for low altitude optical-based SAA will enhance safety for sUAS operations by enabling an on-board aircraft collision avoidance capability of non-cooperative sUAS. Low-altitude optical based SAA is a challenging environment with high image clutter and texture, varying climate conditions, differing intruder aircraft dimensions, and differing intruder aircraft maneuverability. To this end, this work presents a novel sUAS detect and track pipeline using higher pixel-per-degree optics to improve detect and track range of sUAS and GA aircraft.

III. Experimeintal Design

A. *Aircraft*

Two flight scenarios are evaluated in this work: 1) multirotor ownship vs a fixed wing sUAS 2) multirotor ownship vs. GA plane. The aircraft are shown in Fig. 1 through Fig. 3 and their specifications are shown in Table I.

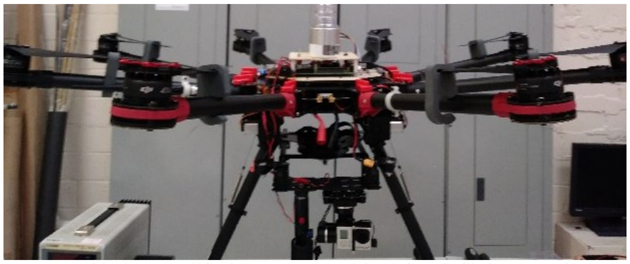

Fig. 1. Ownship is a modified DJI S1000 with pixhawk flight controller.

Fig. 2. N533, Intruder Tempest.

Fig. 3. Intruder SR22.

TABLE I: Aircraft Manufacture and Dimensions

Aircraft	**Make / Manufacture**	**Length x Wingspan x Height (meter)**
ownship	DJI S1000	~1 x 1 x 1
N533NU	UASUSA Tempest	1 x 3.2 x 0.3
NASA601NA	Cirrus SR22	7.9 x 11.7 x 2.7

B. Flight Paths

Aerial images of the flight paths were generated using Google Earth [11] and are shown in Fig. 4 through Fig. 7. The Tempest fixed-wing sUAS, flew in the high slot relative to the ownship with an altitude difference of 30 meters. The tempest speed was approximately twice that of ownship. Table II presents the details on the flight characteristics. The analysis for the tracking algorithm began when the intruder aircraft reached an altitude of 24 meters.

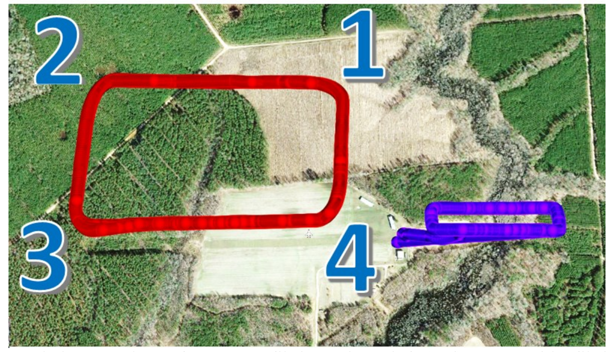

Fig.4. Bird's eye view of Tempest flight path on left (red) vs ownship flight path on right (purple). The tempest flight path follows the 1-2-3-4 pattern for a head-on like scenario.

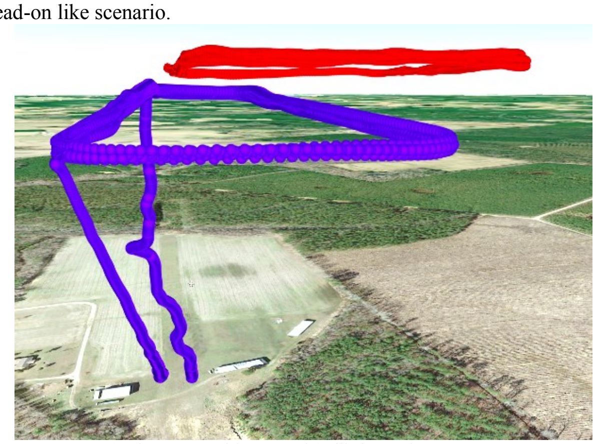

Fig. 5. Tempest (red) vs ownship (purple) flight path.

TABLE II: Tempest vs ownship flight summary.

	Average GPS Altitude (meters)	**Average GPS Speed (meters/second)**
Tempest at flight test altitude	255	16.6
ownship at flight test altitude	225	8.6
ownship including ascent altitudes over 24 meters	192	7.0

The SR22 flew in the high slot relative to the ownship with an altitude difference of 180 meters. The SR22 speed was approximately eighteen times that of ownship. Table III presents the details on the flight characteristics.

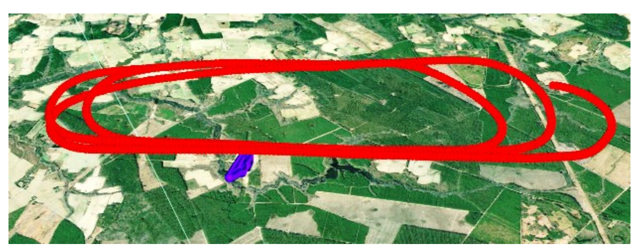

Fig. 6. Bird's eye of SR22 (red) vs ownship (purple). Note the GA plane flies a much larger circuit.

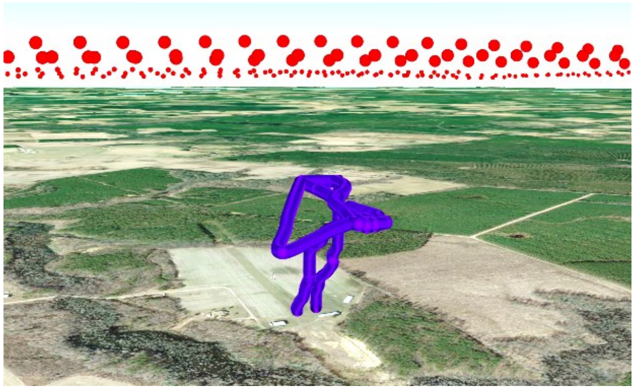

Fig. 7. Ownship (purple) vs SR22 (red). Note a higher altitude separation for safety.

TABLE III: SR22 vs ownship flight summary.

	Average GPS Altitude (meters)	Average GPS Speed (meters/second)
SR22 at flight test altitude	408	57.0
ownship at flight test altitude	227	3.0
ownship including ascent altitudes over 24 meters	202	2.7

C. Camera

One action camera was mounted to ownship for this work and retrofitted with a 41° Field of View lens. The number of pixels per meter for given ranges is shown in Table IV with a comparison to the NASA 2015 AP Hill camera configuration (FOV 170 degrees). The rolling-shutter of the action camera degraded image-quality in the vibration environment of a flying motorcopter.

TABLE IV: Number of pixels per meter for given range

	Horizontal Field of View	
	170°	41°
Separation Distance (ft)	pixels per meter	pixels per meter
2500	2.46	6.77
1500	4.11	11.29
1000	6.16	16.93
750	8.21	22.57
500	12.32	33.86
400	15.40	42.32
200	30.79	84.64
100	61.59	169.28
25	246.35	677.11

IV. Detect and Track Pipeline

An overview of the detect and track pipeline is shown in Fig. 8. The first step is to segment the above-horizon-region because a different set detection and tracking parameters is needed for the below-the-horizon region due to the presence of high-textured terrain and ground clutter. Next, two different types of image-detectors were used to detect the sUAS. These detectors tend to produce high false positive rates for small targets in image space. A Kalman-filtering approach is used to establish tracks from these detections and a portion of these tracks are established as "Good Tracks" should they satisfy image-space movement and age thresholds. The image-processing methods are implemented using the OpenCV library [12] and the tracking algorithm methods are implemented using MATLAB [13].

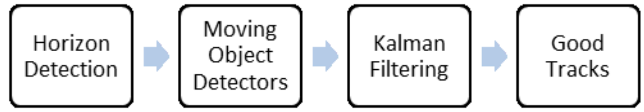

Fig. 8. Detect and Track Pipeline

A. Horizon Detection

The horizon detector, shown in Fig. 9, assumes that the longest contour output is the horizon, which appears to be a reasonable assumption for low altitude flights at this test-site. The Otsu's thresholding method and Canny Edge detector are utilized to extract the horizon contour [14] [15].

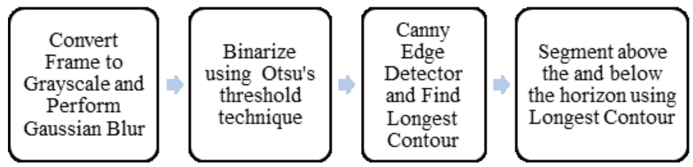

Fig. 9. Horizon Detector

B. Moving Object Detectors

This section presents the algorithm details for image-differencing and the morphological sUAS detection technique.

1) Image Differencing Detector

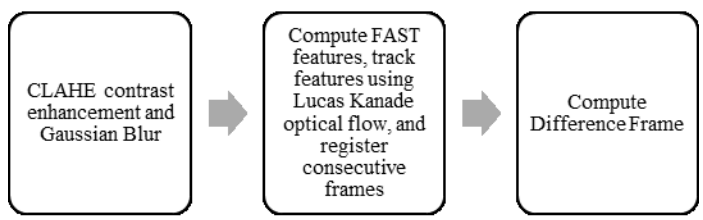

Fig. 10. Image Differencing Based Detection

The image differencing method subtracts the previous frame from the current frame as shown in Fig 10. First, image keypoints are detected using the FAST feature detector for the previous frame for the first image [16]. Next, the previous keypoints are found in the current frame using a pyramidal implementation of Lucas Kanade Optical Flow [17]. The homography between the previous frame and the current frame is computed using the perspective transformation approximation:

$$s\begin{bmatrix} x_i^{prev} \\ y_i^{prev} \\ z_i^{prev} \end{bmatrix} \sim H \begin{bmatrix} x_i^{current} \\ y_i^{current} \\ z_i^{current} \end{bmatrix} \qquad (1)$$

where, s is a scaling vector, x, y, z are position coordinates, and H is the transformation matrix, which is minimized using back-projection error:

$$\sum\left(x_i^{prev} - \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}}\right)^2 + \left(y_i^{prev} - \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}}\right)^2 \quad (2)$$

Using the homography matrix, a perspective transformation for the previous frame is performed. The absolute difference is computed using the current frame and the transformed previous frame. As the difference values in consecutive frames tend to be about 10 to 15 in pixel intensity, a binary image generated using a threshold intensity of 10 is only used for the visualization in Fig. 11.

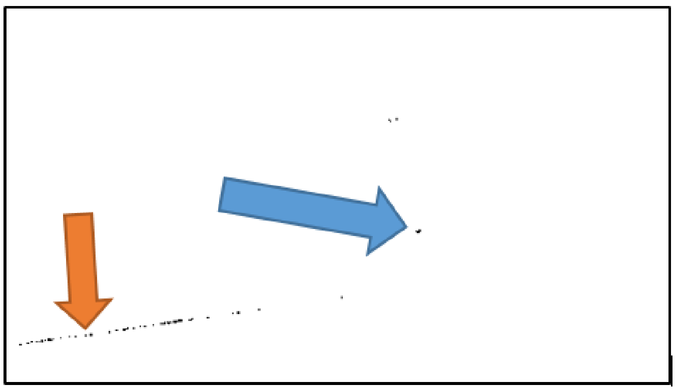

Fig 11. Sample Binarized Difference Frame – inverted for display purposes. The intruder sUAS is slightly right and down from the center pixel as indicated by blue arrow. An artifact is generated from the horizon segmentation and image differencing process as indicated by the orange arrow. A dynamically thresholding FAST Feature method extracts only the intruder from the difference frame.

The difference detections are determined using a dynamic threshold for Fast Feature detection. The FAST detector threshold is the contrast requirement for a given pixel relative to a surrounding circle of pixels. The threshold value is initialized as 10 and incremented until the number of keypoints is less than a predetermined value. For this work, it was assumed that there are no more than 5 intruders. Generally, the horizon line artifacts are removed by the Fast Feature thresholding, however, an algorithm may be developed to mitigate for the differences in horizon location between consecutive frames.

2) Morphological sUAS Detector

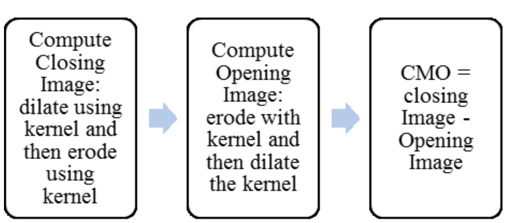

Fig. 12. Morphological Based Detection

The morphological sUAS detector is shown in Fig 12. The first step is to compute the closing image, next the opening image is computed, and finally the difference image is computed by subtracting the closing image from the open image. The following paragraphs detail this calculation.

The outputs for the close-minus-open figure are shown in Fig. 13 through Fig. 15. The opening filtering helps remove noise from the image. For this work, a cross shaped kernel is used:

$$K = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3)$$

The closing image is computed by first dilating using the cross shaped kernel and then eroding with the same kernel. The opening image is computed by first eroding with the kernel and then dilating with kernel. This morphological filtering approach works well for objects on low textured backgrounds, e.g. clear sky. The close-minus-open process produces good contrast for the sUAS from the background as shown in Fig. 15.

Fig. 13. The closing process first dilates using the kernel (a plus shaped kernel in this application) followed by an erosion.

Fig. 14. The opening process first erodes using the kernel (same plus shaped kernel) followed by a dilation.

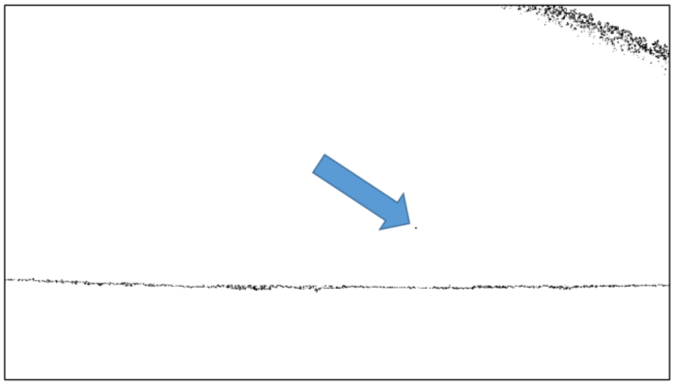

Fig. 15. The close-minus-open filter requires subtracting Fig.13 from Fig. 14 to yield the above result, which has been enhanced for visualization purposes. The intruder sUAS as indicated by the blue arrow. The horizon line is apparent as the jagged edges are detected from the close-minus-open process. The halo effect in the upper-right hand corner may be a result of MPEG compression artifact.

3) Sample Detections

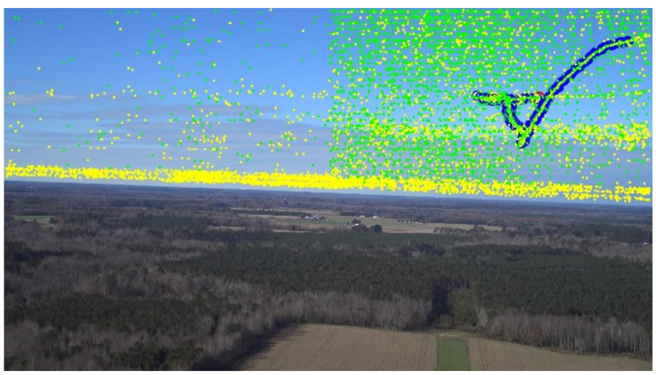

Fig. 16. Green is the morphological detections and yellow is the difference detections. The blue track is generated using the Kalman based tracker.

Sample detections are shown in Fig. 16 where green is the morphological detections and yellow is the difference detections. The blue track is generated using the based Kalman tracker. The higher density of detections to the right may be partially the result of movement from the sUAS track from the right of the screen.

A single detection frame is shown in Fig. 17 with a zoomed result shown in Fig. 18. The morphological detector has two false positives: one is likely the result of a data encoding artifact as shown in Fig. 19 and the other is generated from the edge of a cloud.

Fig. 17. Sample detector result where green is the difference detector and red is the morphological detector.

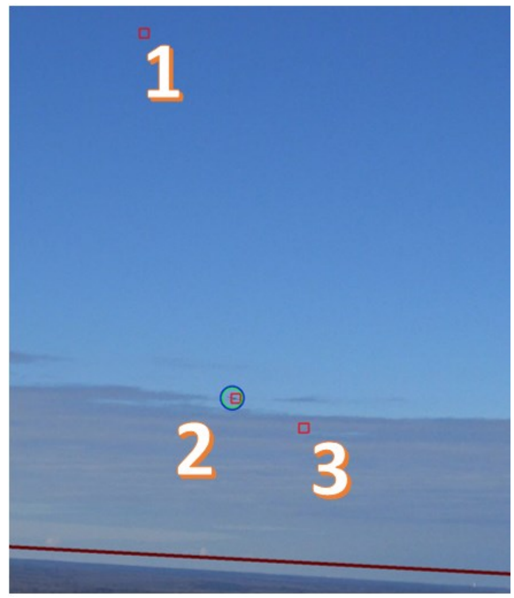

Fig.18. Zoomed in detection result. Note false morphological detection indicated by the number 1, which is likely a result of MJPEG encoding. The number 2 shows the positive detection by both differencing and morphological detections. The number 3 shows a false detection from a gap in the cloud.

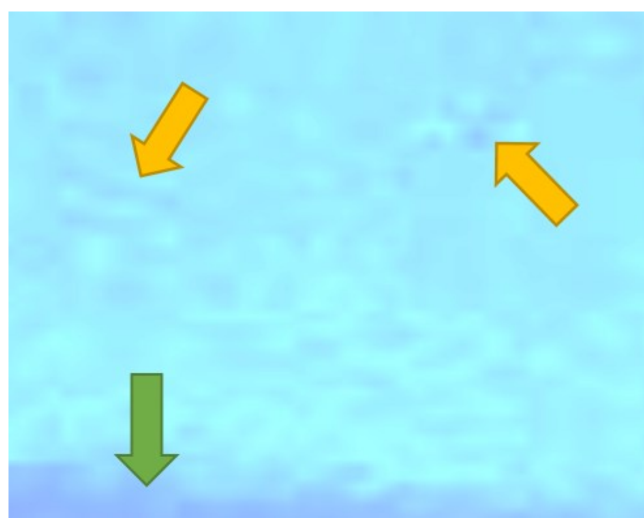

Fig. 19: Sample Zoomed Image. MJPEG encoding generated spatial and temporal artifacts as indicated by the yellow arrows, resulting in false positives for small object detection. The darker region in the bottom is part of a cloud as indicated by the green arrow.

C. Tracking Method

For a given frame, each detection is either assigned to a new track or an existing track by minimizing a negative log likelihood cost function using a variation of Hungarian assignment algorithm [18]. The Kalman filter assumed a constant velocity with empirically determined parameters: an initial estimate uncertainty variance of 10 by 10, motion noise 20 by 20, and measurement noise 20. Each track is described by: the current location, Kalman Filter parameters, age (number of frames since initial detection), total number of frames where the object was detected, and total number of consecutive invisible counts. The Kalman filter updates the predicted location for each existing track. Tracks that are invisible too long are deleted. For this 30 Hz camera configuration, tracks that were invisible more than 5 frames were deleted and a track needed to have at least 30 detections to be considered a "good track".

V. Detection and Tracking Results

The results are presented for the entire videos for both types of encounters and also an in depth analysis of when the intruder sUAS was physically in the frame of the video. Detection of these small targets tended to produce a high false positive rate as initially shown in Fig. 16. Kalman filtering of the detector outputs enables the establishment of good tracks.

A. Image detectors

Detection accuracy is computed by summing the number of true moving object detections and dividing by the total number of detections. The morphological detector was consistently more accurate than the difference detector as shown in Table V. These two small object detectors are often confounded by clouds and may each produce up to 6 detections per frame.

Table V. Detection Results

Encounter	# frames	Diff Detector ACC %	Morph Detector ACC %	Min sep dist (m)	Max sep dist (m)	Mean sep dist (m)	Std sep dist (m)
Ownship vs N533 video	18538	5.2	12.2	267	1225	694	253
Sortie 1	181	10.5	13.2	353	400	375	15
Sortie 2	1513	9.5	16.8	746	1020	938	87
Ownship vs Helicopter	2223	17.5	73.2	-	-	-	-
Sortie 3	1479	16.0	45.8	639	1190	990	174
Sortie 4	1530	7.9	25.3	779	1193	1056	121
Sortie 5	1330	10.5	23.9	851	968	915	29

Detection statistics vs range is shown in Fig. 20 and Fig. 21. The parameters of these detectors are tuned to increase the likelihood of detection of these small target intruder aircraft while sacrificing low false positive rates. Separation distances of ~400 to 1000 meters include detections for the approach leg of the encounter (between waypoints 3 and 4 in Fig. 4). Sample false positive detections are shown in Fig. 22. Many of these detections occurred near the sensor limit, therefore, a subset of the false detections may be birds – for instance the top right image of Fig. 22 appears to have a spec that cannot be confirmed as a bird or a compression encoding artifact.

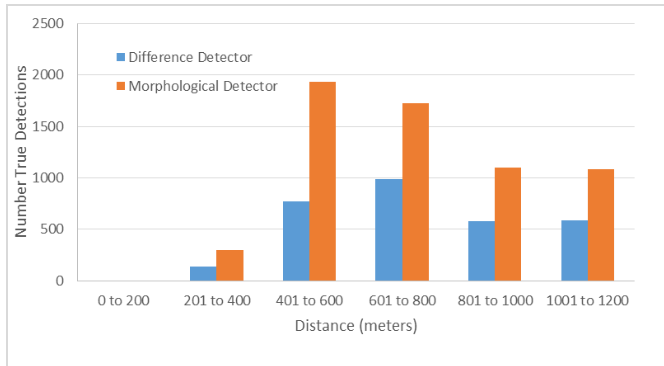

Fig.20: Number of Detections vs Range.

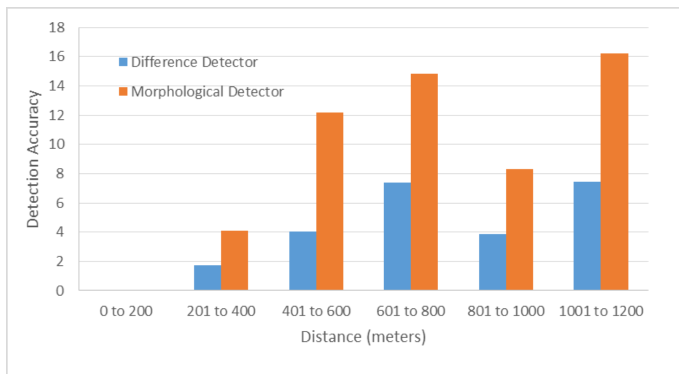

Fig. 21. Detection Accuracy vs Range.

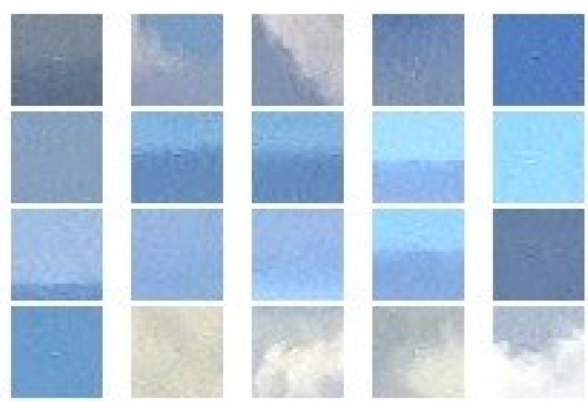

Fig. 22. Sample False Positives from image detectors.

Sample detections for the far leg of the Tempest Flight Path (between waypoints 2 and 3) are shown in Fig. 23, the approach leg (between waypoints 3 and 4) in Fig. 24, and the front leg in Fig 25.

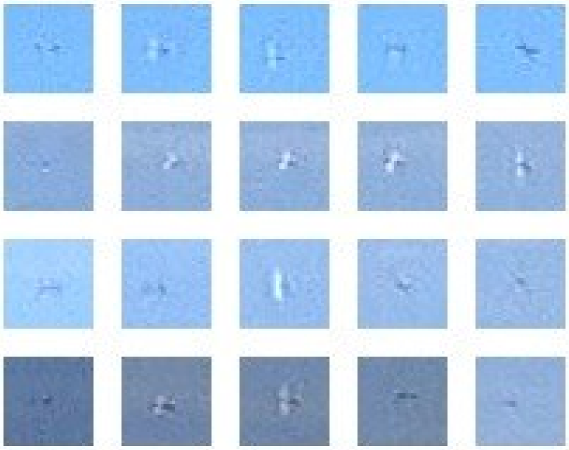

Fig. 23. Sample N533 detections at the Back Leg of the flight path between waypoints 2 and 3.

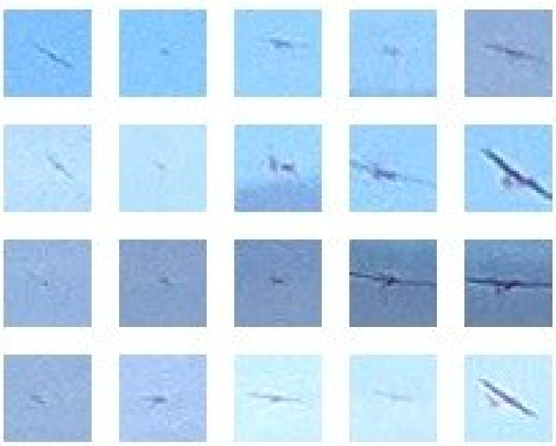

Fig. 24. Sample N533 detections at the Approach Leg of the flight path between waypoints 3 and 4.

Fig. 25. Sample N533 detections at the Front Leg of the flight path between waypoints 4 and 1.

B. Tracking

The tracking results are superimposed on the first image of the sortie and are shown in Fig. 26 through Fig. 32, where the red circles shows the initial establishment of the track, the blue circles indicate a ground truth confirmed track label, and the yellow circles indicate a tracker predicted location that does have a labeled ground-truth. These tracker figures do not take in account the motion of the camera and that the ownship is performing the box patterns in Fig. 4 and Fig 6.

The ownship ascended during sortie 1 (Fig. 26) and just caught the end of the approach leg and turn of intruder N534. In sortie 2 (Fig. 27), the tracker establishes a good track after the intruder enters from the right, tracks during the turn between far leg and the approach leg, tracking is intermittent at the far-end of the approach leg due to low contrast background of the cloudy background, and tracking solid at the near-end of the approach leg and the front leg. Sortie 3 (Fig. 28) is a multi-vehicle with a bonus unexpected helicopter. The helicopter enters the video first from the left with relatively constant image speed with its flight path is tracked in Fig 31. Tracking is not continuous in Sortie 3 partially as a result of a change in light balancing in the imagery, which degraded sUAS contrast relative to the cloudy background. Another possibly impacting factor to the tracking is the small size of the target in the imagery and the vibration environment for the rolling shutter. In Sortie 4 (Fig. 29), the sUAS in tracked when it enters from the right in the darker cloudy region, tracking is lost after the turn between the back leg and the approach leg, tracking resumes again later in the approach leg and is nearly through the exit at the front leg with the exception of momentary loss during the transition between the darker and lighter cloud backgrounds. In sortie 5 (Fig. 30), the sUAS is tracked after entering from the right into the imagery, is lost after the turn between the back leg and the approach leg, tracking resumes in the approach leg and is near continuous through the exit at the front leg with the exception of a lost track during a transition between sky to cloud. The helicopter was tracked 99% of the frames during its sortie. Characteristics that made the helicopter easy to track were: large cross-sectional image area, high contrast black airframe (unlike white wings of either N533 or SR22) relative to the cloud cover, and constant image speed and heading. In Sortie 6 (Fig. 32), the SR22 had a good image speed, however, tracking was intermittent to due low-contrast busy cloud-covered background. The cloudy background generated more false detections as result of many more edges around the clouds. Dynamically tuning the parameters for morphological and difference detectors may help detect the SR22 on low-contrast backgrounds.

Fig. 26. sUAS sortie 1.

Fig. 27. sUAS sortie 2.

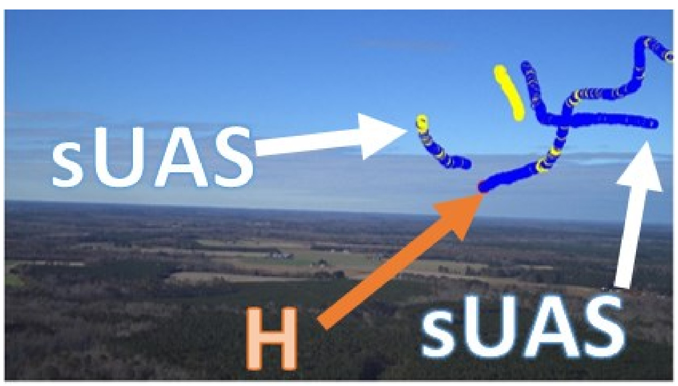

Fig. 28. sUAS sortie 3. The H label indicates the helicopter track and the sUAS arrows show the beginning of the sUAS tracks

Fig. 29. .sUAS sortie 4.

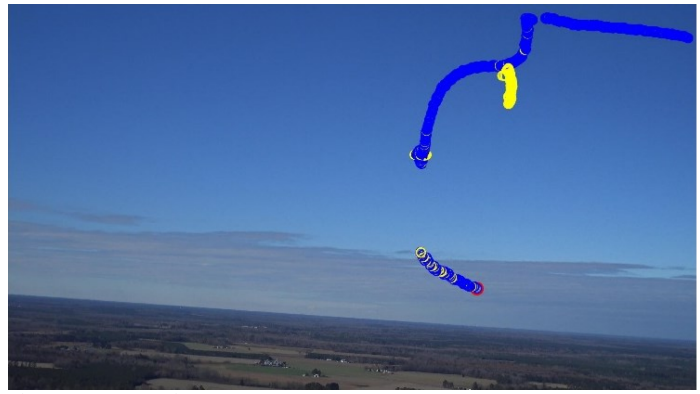

Fig. 30. sUAS sortie 5.

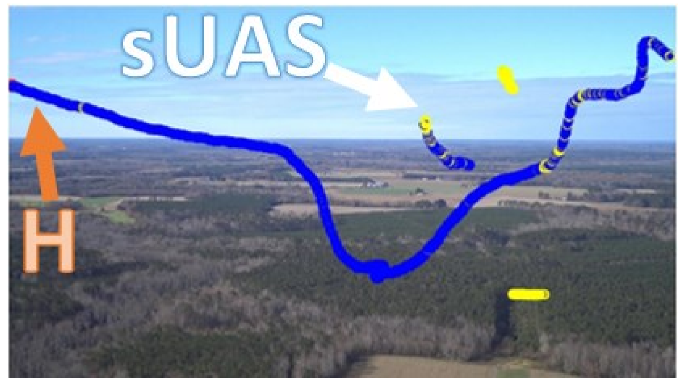

Fig. 31. sUAS vs helicopter. Note that the helicopter held roughly the same altitude and that the curvature of the helicopter tracking line is a result of the camera carrier's movement.

Fig. 32. Sortie 6: sUAS vs SR22.

The tracker returned only true tracks for the intruder aircraft for 5 of 6 sorties as shown in Table VI. The correctly tracked frame rate suffered most when the sUAS entered low-contrast cloud regions, where in many cases the sUAS was not visually present in the imagery to detector or human eye. Overall, the tracking method returned a good track rate of 0.71 for the ownship vs N533 video, which was recorded on a partly cloudy day. The SR22 flight included greater cloud coverage, which generated a lower good track rate for the SR22 video. A planned implementation of a static object filter should help suppress false detections from edges of clouds and thus likely improve the good track rate in cloudy conditions.

Table VI: sUAS Tracking Results

Encounter	# frames	Seconds	Good Track Rate	% correctly tracked frames
ownhsip vs N534 video	18538	617.9	0.71	NA
Sortie 1	181	6.0	1.00	33.7
Sortie 2	1513	50.4	1.00	52.0
ownship vs Helicopter Sortie	2223	74.1	1.00	99.9
Sortie 3	1479	49.3	0.67	33.9
Sortie 4	1530	51.0	1.00	60.5
Sortie 5	1330	44.3	1.00	66.2
ownship vs SR22 video	18235	607.8	0.29	NA
Sortie 6	693	23.1	1.00	83.0

VI. CONCLUSION

A sUAS far-field sUAS detect and tracking pipeline was presented in this work. The tracker performed best when the intruder aircraft had faster image-speed, high-contrast relative to the background, and larger cross-sectional areas. Combining the two types of detectors improved the tracking result, thus an optical-based non-cooperative aircraft system will benefit from multiple types of detectors with dynamic detecting and tracking methods to overcome the challenges of varying backgrounds, intruder aircraft contrast, speed, and heading.

VII. FUTURE WORK

Future plans include integrating the attitude information from the flight controller for computing the horizon in the imagery, transforming 2D image detections to spherical coordinates using attitude to improve tracking results by taking into account the movement of the camera and vehicle, and implementing a static object filter to suppress non-moving false positives from cloud edges. These improvements should help with detection and tracking of sUAS below the horizon, where the background will have high-texture surfaces such as trees and other ground-clutter.

ACKNOWLEDGMENT

The authors would like to thank sUAS pilots: Kyle Smalling from the National Institute of Aerospace (NIA), Dave Bradley (NIA), and Nick Rymer (NIA); sUAS ground station operators Swee Balachandran (NIA), Brendan Duffy (NIA), and Mike Logan (NASA); range safety officer Jeff Hill (NASA); Henry Holbrook for labelling SR22 ground truth data.

REFERENCES

[1] Federal Aviation Administration, "Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Concept," 2018.

[2] Federal Aviation Administration , "FAA Aerospace Forecast: Fiscal Years 2018-2038," 2017.

[3] G. Fasano, D. Accardo, A. Tirri and A. Moccia, "Experimental Analysis of Onboard Non-Cooperative Sense and Avoid Solutions Based on Radar , Optical Sensors, and Data Fusion," *IEEE A&E SYSTEMS MAGAZINE,* vol. 31, no. 7, pp. 6-14, 2016.

[4] M. Wilson, D. Ryan, D. Bratanov, A. Wainwright, J. Ford, L. Cork and M. Brouckaert, "Flight Test and Evaluation of a Prototype Sense and Avoid System Onboard a ScanEagle Unmanned Aircraft," *IEEE A&E Systems Magazine ,* vol. 10, no. 1109, pp. 6-15, 2016.

[5] J. Lai, J. Ford, L. Mejias and P. O'Shea, "Characterization of Sky-region Morphological-temporal Airborne Collision Detection," *Journal of Field Robotics,* vol. 30, no. 2, pp. 171-193, 2013.

[6] A. Nussberger, H. Grabner and L. Van Gool, "Aerial Object Tracking from an Airborne Platform*," in *Unmanned Aircraft Systems (ICUAS)*, Orlando, 2014.

[7] D. Ye, J. Li, Q. Chen, J. Wachs and C. Bouman, "Deep Learning for Moving Object Detection and Tracking from a Single Camera in Unmanned Aerial Vehicles (UAVs)," *Society for Imaging Science and Technology,* vol. 466, pp. 1-6, 2018.

[8] C. Dolph, M. Logan, L. Glaab, T. Vranas, R. McSwain, Z. Johns and K. Severance, "Sense and Avoid for Small Unmanned Aircraft Systems," in *AIAA Information Systems-AIAA Infotech@ Aerospace*, Grapevine, 2017.

[9] H. Seviil, A. Dogan, K. Subbarao and B. Huff, "Evaluation of Extant Computer Vision Techniques for Detecting," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, 2017.

[10] "Small Aircraft Flight Encounters (SAFE) Dataset Repository," 2019. [Online]. Available: https://sites.google.com/vt.edu/safe-repository.

[11] Google , "Google Earth Pro," 2017.

[12] OpenCV, Open Source Computer Vision, 13 February 2018. [Online]. Available: https://docs.opencv.org/3.4.1/index.html.

[13] MathWorks, "MATLAB 2018a," Natick, 2018.

[14] N. Otsu, "A Treshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 9, no. 1, pp. 62-66, 1979.

[15] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal.. Machine Intell.,* vol. 8, pp. 679-698, 1986.

[16] E. Rosten and T. Drummond, "Machine learning for highspeed corner detection," in *European Conference on Computer Vision*, Graz, 2006.

[17] J. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," Intel Corporation, 2001.

[18] J. Munkres, "Algorithms for Assignment and Transportation Problems," *Journal of the Society for Industrial and Applied Mathematics,* vol. 5, no. 1, 1957.

[19] F. A. Administration, "PART 107—SMALL UNMANNED AIRCRAFT SYSTEMS," Electronic Code of Federal Regulations, 2 March 2017. [Online].Available: http://www.ecfr.gov/cgi-bin/text-idx?SID=e331c2fe611df1717386d29eee38b000&mc=true&node=pt14.2.107&rgn=div5. [Accessed 4 March 2017].

[20] G. Bradski and A. Kaehler, "Learning OPENCV: Computer vision with the OpenCV Library," 2008.