Sense and Avoid for Small Unmanned Aircraft Systems

Chester V. Dolph[1], Michael J. Logan[2], Lou J. Glaab[3], Thomas L. Vranas[4], Robert G. McSwain[5], Zachary R. Johns[6], Kurt Severance[7]

The ability for small Unmanned Aircraft Systems (sUAS) to safely operate beyond line of sight is of great interest to consumers, businesses, and scientific research. In this work, we investigate Sense and Avoid (SAA) algorithms for sUAS encounters using three 4k cameras for separation distances between 200m and 2000m. Video is recorded of different sUAS platforms designed to appear similar to expected air traffic, under varying weather conditions and flight encounter scenarios. University partners and NASA both developed SAA methods presented in this report. This report is organized as follows: Introduction, Experimental Design, Description of video and GPS data, NASA Methods and Results, University Partner Results, Recommendations, and Conclusions.

I.   Introduction

RECENT proliferation of sUAS necessitates improved autonomous SAA algorithms for Beyond Visual Line of Sight (BVLOS) operations, where the sUAS needs the ability to autonomously fly without interfering with commercial flights, general aviation aircraft, and other sUAS. NASA is engaged in the Unmanned Aircraft Systems Integration in the National Airspace System (UAS in the NAS) project, where the goal is to overcome the technical and operational obstacles for conventional safe UAS flights in the NAS. While the majority of the project is focused on large UAS interacting with commercial air traffic, this portion of the project investigates the integration of sUAS in the NAS. The motivation of our work is to design and build an object detection system that would allow autonomous sUAS to operate beyond visual line of sight without the need for Ground-Based Sense-and-Avoid (GBSAA) systems such as ground radars. The first of two flight tests have been completed. The results from our first flight test are presented in this abstract and the second set of flight tests is currently being planned.

This section describes the requirements of SAA systems for integrating sUAS into the National Airspace (NAS). The video from this work may be used for the development of SAA on embedded systems, thus this video database is a resource for developing systems that overcome the technical barriers of integrating sUAS into the NAS.

There are four requirements of SAA systems: Separation Assurance, Collision Avoidance, Sensing Requirements, and Avoidance Requirements[1]. A minimum operating distance between sUAS is needed to operate safely and meet the separation requirement. The separation distance will vary depending on airspace class, flight phase, and other flight conditions. sUAS should always satisfy the separation distance; however, a collision avoidance system should act as a failsafe should the separation distance be violated. The collision avoidance system may alter course to prevent midair collisions autonomously (sUAS embedded system executes SAA flight change), semi-autonomously (sUAS alters course based on radio communication from the Ground Control Station (GCS)), Air Traffic Management (central station controlling flight paths), or other operational procedure (fly up and to the left). We are investigating a solution for the collision avoidance of non-cooperative aircraft (i.e. those not using a transponder such as Automatic Dependent Surveillance-Broadcast (ADS-B)) with our optical SAA platform.

The sensing requirement concerns the detectability of all possible intruders that a sUAS may encounter during a flight. Variables influencing the sensing requirement include: sensor range, orientation, resolution, instantaneous field of view, total field of regard, sampling rate, and sensor type (e.g. optical, radar, or LIDAR). Optical sensors may fail the sensing requirement during inclement weather while radar sensors are largely uninfluenced by clouds or

---

[1] Research Engineer, National Institute of Aerospace, 100 Exploration Way, Hampton, VA
[2] Head of Small Unmanned Aerial Vehicle Laboratory, NASA Langley Research Center, Hampton, VA
[3] Assistant Branch Head, ASEB at NASA Langley MS 238
[4] Aerospace Research Technologist, National Institute of Aerospace, 100 Exploration Way, Hampton, VA
[5] Aerospace Engineer, ASEB at NASA Langley MS 238, Member AIAA
[6] Pilot, National Institute of Aerospace, 100 Exploration Way, Hampton, VA
[7] Computer Engineer NASA Langley Research Center, Hampton, VA

precipitation. The avoidance requirement describes how the sUAS should respond after detection of another sUAS. Possible solutions are change in speed, altitude, and heading. Our focus is on investigating non-cooperative SAA between sUAS and manned aircraft, which is extensible to SAA between sUAS and sUAS. The SAA algorithms we investigate may be implemented using state of the art embedded systems, assuming a suitable algorithm can be determined.

II.   Experimental Design

This section describes the sUAS with cameras, intruder sUAS, and the flights flown in this project.

A. Camera Platform

Three 4k Sony Action Cameras were mounted to a fixed wing sUAS dubbed the Mig Camera Carrier (MigCC) shown in Figure 1. One camera is mounted behind the propeller while two more cameras are located on the wingtips, all of which are oriented in the forward direction of the sUAS. The use of multiple cameras enables depth calculations to be performed using stereo vision. Measurements of the MigCC and relative camera geometry are presented in Table 1.
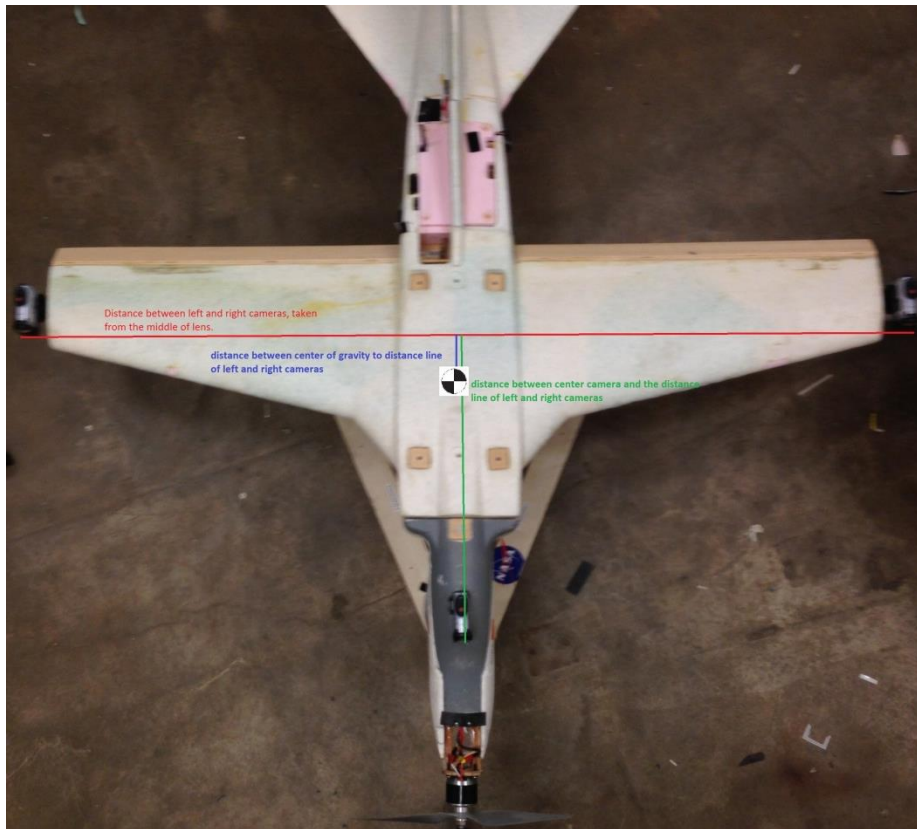

Figure 1. MigCC platform. *The left and right cameras are located on the wing tips and the center camera is positioned just behind the propeller.*

Table 1. Geometry of cameras and the MigCC.

Line Segment	Distance	Measurement (inches)	Line color
D1	Left and Right Cameras	69 and 3/16	Red
D2	Center Camera Lens to D1	24 and 1/8	Green
D3	Center of gravity to D1	3 and 3/16	Blue

B. Intruder sUAS

We collected video data for three intruder sUAS: Y-6, Bixler, and a 21% scale model of a Cirrus SR22 general aviation aircraft. The intruder sUAS are shown with the MigCC in Figure 2.

Figure 2. sUAS encounter fleet. *The sUAS are shown from left to right: MigCC, Y6, Bixler, and SR22.*

C. Flight Tests

We completed 10 flight tests over 3.5 days that varied in weather from sunny, partly cloudy, and low overcast. Three elevations were explored for our flight test: MigCC and intruder aircraft at the same altitude, intruder aircraft at a higher altitude than MigCC, and intruder aircraft at a lower altitude. During the experiment, the sUAS were either controlled by a ground control station or pilot. The distance between the MigCC and the intruder sUAS ranged from 200 to 2000 meters in flight.

III.   Description of video and GPS data

This section describes the three cases of video where: the intruder is not visible in the video feed, is present but lacks defined geometry, and exhibits recognizable sUAS geometric features. There are 4.6 hours of video data and 1.5 hours where the MigCC is oriented in the direction of the intruder sUAS. Table 2 presents the recorded video with regard to weather condition, flight length, and the total time MiggCC is oriented towards the intruder (where the intruder should be present in the video).

Table 2. Flight Weather and Time

Weather Condition	Number of Flights	Total Time MigCC in Air (hours)	Total Time MiggCC Oriented Towards Intruder sUAS (hours)
Sunny	4	1.98	0.70
Partly Cloudy	2	0.95	0.33
Low Overcast	4	1.73	0.45
Total Number of Flights	10	4.66	1.48

A. Intruder is not visible in video feed

The sUAS may not be visible due to following complications: weather, terrain, horizon, or insufficient movement within the video frame. The weather can eliminate sUAS sightings or cause intermittent sightings due to lack of contrast between sUAS and background (not due to atmospheric visibility limitations— the minimum visibility for our flight operations is 5 miles). The scale SR22 sightings largely occur on the sunny days and are occasionally visible on partly cloudy days, with few sightings on low overcast days. On partly cloudy days, the SR22 is intermittently visible in video due to flying in front of clouds where it has varying levels of contrast relative to the white and grey backgrounds.

The horizon resembles a blurred line where the sUAS would blend into the gray, light-blue, and dark colors. One of the trajectories shows the sUAS ascending to a higher altitude. During this flight it is visible above the tree line, vanished within the horizon, and then reappears later in the sky.

B. Intruder resembles a spec devoid of well-defined geometry

The scale SR22 resembles a spec the majority of the time it is visible. Figure 1 shows one instance where the spec is enlarged for visibility. Visual detection is achieved by a human by looking at the right part of the frame at the right time to see the spec move across it. Due to the small number of pixels of the sUAS, partially due to the large field of view, it often blends in with the trees. The texture of the wooded areas resulted in a blinking sUAS as it moves from darker to lighter areas of the trees. The spec typically yields only one keypoint for a given feature detector, which is lost as the sUAS flew across texture changes in the trees.

Figure 1: Example SPEC sUAS. *The SR22 is approximately 5 by 5 pixels without robust geometry and significant contrast from the trees.*

C. Intruder exhibits expected shape

The intruder aircraft exhibited the most robust geometry while banking in front of the MigCC as shown in Figure 2. Robust geometry implies that shape of the sUAS in the image resembles the scale SR22.

Figure 2: Example of well-defined sUAS. *The SR22 has the expected geometry with about 14 by 14 pixels at a separation distance of approximately 300m.*

D. GPS data

The video data is integrated with the onboard GPS data to provide MigCC and intruder location analysis after the flights are completed. Figure 3 shows the GPS locations of the intruders and MigCC with a separation distance plot and Figure 4 shows a sample GPS track of our flight tests. Matching the 4k video with the GPS data enables the determination of when an intruder should be more easily detected because it is closer to the MigCC and when it is more difficult to detect because it is further away from MigCC.
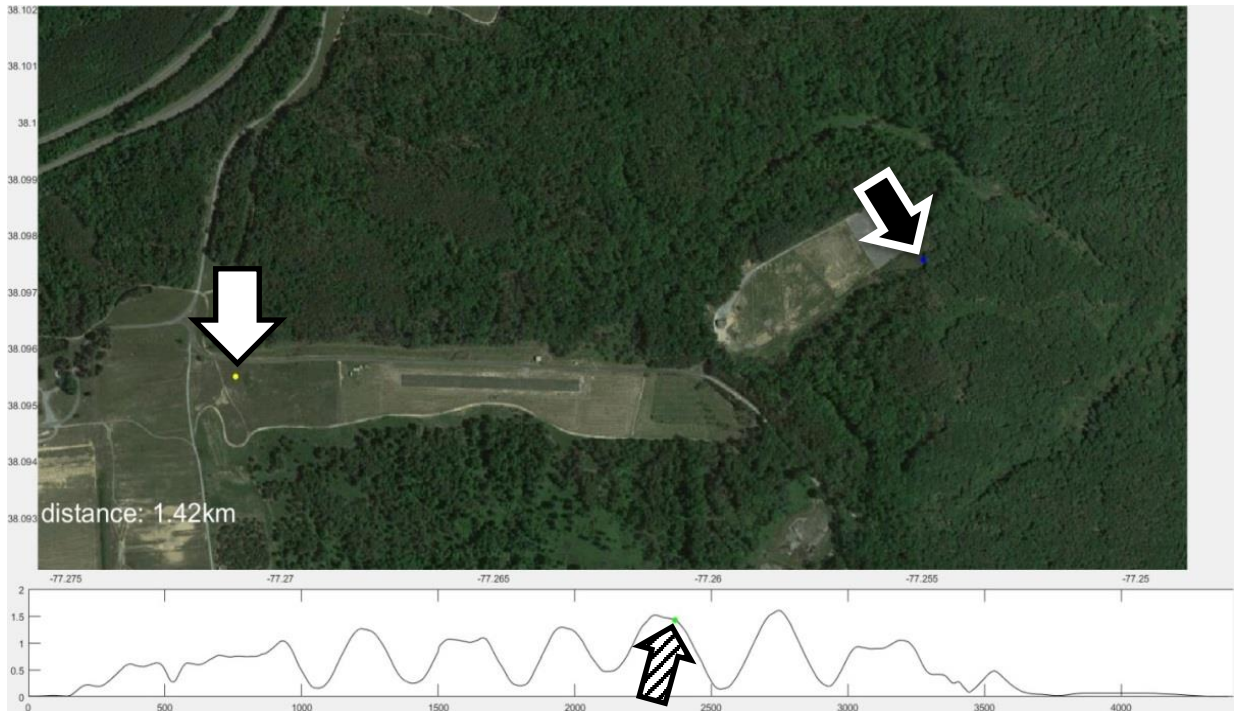

Figure 3: GPS data and distance calculations. *The top plot is a Google GPS image of the flight area where the blue dot is the MigCC (black filled arrow) and the yellow dot is the scale SR22 (white filled arrow). The bottom plot is the distance between the sUAS vs time. The black and white patterned arrow indicates the location and separation distance of the sUAS shown in the Google GPS image.*

Figure 4: Flight Paths for MigCC vs SR22. *The MigCC is on the right with the yellow elevation lines and the SR22 is on the left with the orange vertical lines. Note that the SR22 is in pilot in command mode and the MigCC is in AutoNav mode. The SR22 violates the buffer zone only on takeoff and the MigCC takes off only after the SR22 has reached its established altitude.*

IV.   NASA Methods and Results

Three object detection and tracking methodologies are applied to the collected data, one based on SURF[2], another based on Lucas-Kanade optical flow, and finally one that uses optical flow for consecutive frame registration and FAST features for tracking. SURF identifies specific objects using a pre-computed database of SURF descriptors. The Lucas-Kanade optical flow algorithm uses spatial-temporal relationships from the video to extract pixel movement for tracking. In general, SURF method is most useful for determining an exact object (i.e. Cirrus SR22 or Y-6), while optical flow algorithms are blob trackers that may not necessarily discern between objects (sUAS vs birds) at their most basic implementation. The FAST feature pipeline computes the difference frame for two consecutive frames and achieves positive detection based on the principle that the intruder sUAS is moving quickly relative to the foreground of the image. The SAA algorithms are implemented on a desktop using OpenCV 3.0[3]. Figure 5 below shows the proposed pipeline with the elements common to both SAA algorithms.

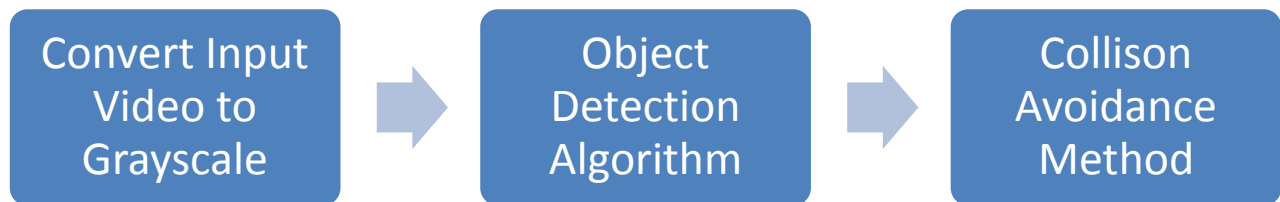| Convert Input Video to Grayscale	→	Object Detection Algorithm	→	Collison Avoidance Method

Figure 5: Proposed object detection algorithm.

The first step is to convert the input video to grayscale because luminance is more important than chrominance for distinguishing visual features. Further grayscale images decreases the amount of data that needs to be processed from three color channels to a single grayscale channel. The data reduction is important for embedded systems, which have more limited resources than high performance tower computers. The next two steps in the SAA pipeline: the object detection searches for intruder sUAS and triggers collision avoidance system if needed. The object detection algorithms are discussed below.

A. SURF method

The SURF method is an improved and optimized feature extracting method derived from SIFT[4]. An overview of the algorithm is in Figure 6.

Interest Point Detection	→	Local Neighborhood Orientation Computations	→	Matching

Figure 6: SURF pipeline.

The steps of the SURF object detection pipeline are below:

1. Interest Point Detection: SURF uses a blob detector based on the Hessian matrix of the convolution of the image with a second derivative of a Gaussian. The Gaussian kernel is approximated to reduce the number of computations. The Hessian matrix represents the amount of change in a local area and consequently, a large determinant of the Hessian matrix indicates a large change in the image information. The process is repeated to achieve scale invariance at the scales: 9 x 9, 15 x 15, 21 x 21, and 27 x 27 pixels. Using multiple spatial scales ensures that the keypoints are stable, which improves the robustness of the algorithm and also enables detection to occur at varying distances between the camera and the object.

2. Local Neighborhood Orientation Computations: Rotation invariance is achieved by calculating the orientation of each of 4 by 4 pixel subregions surrounding the Interest Points. The Haar-wavelet responses are computed in the horizontal and vertical directions for the subregions surrounding each Interest Point. The dominant orientation is assigned based on the sum of the response vectors of the Haar-wavelets for a given

subregion surrounding the point of interest. A descriptor for each point of interest is calculated using 16 subregions surrounding the point of interest. Each subregion is described by 8 values: the sum of the positive horizontal Haar-wavelet response, the sum of the positive vertical Haar-wavelet response, the absolute value of the positive horizontal Haar-wavelet response, the absolute value of the positive vertical Haar-wavelet response, and four corresponding negative wavelet calculations for each of the previous wavelet responses. In this way, each point of interest has a descriptor that contains 128 values by multiplying the number of subregions (16) by the number of orientation descriptors (8).

3. Matching: Each image will have a descriptor matrix N by 128 descriptor image, where N is the number of points of interest. The descriptor matrix of a given video frame is checked with a reference library of precomputed descriptors of objects using Euclidean Norm.

The SURF algorithm is ineffective for sUAS detection on this data because SURF needs multiple robust keypoints to identify objects. The scale SR22 resembles a morphing spec the majority of the time it is visible, which does not satisfy the distinctive feature requirements of significant geometric convexity or concavity for SURF[4]. We previously used the SURF algorithm to detect a manned aircraft using an 80 by 85 pixel reference image from video recorded in preparation of this flight test. In SURF, several robust keypoints are needed for object detection – too few pixels (at most ~ 15 by 15 pixels of sUAS in this SAA dataset) are available for keypoint detection of inputs for SURF sUAS detection at these separation distances with these cameras and lenses. Further, the 3D changing of orientation of the sUAS results in vastly different 2D images, thus matching keypoints is confounded by the common changes of orientation associated with flying. The texture of the wooded areas results in a blinking sUAS as it moves from darker to lighter areas of the trees further decreasing the ability to find edge points. The spec typically yields only one keypoint, which is lost as the sUAS flew across texture changes in the trees.

B. Lucas-Kanade Optical Flow Tracking

Initialize Shi-Tomasi Corner Points with Subpixel Refinement → Lucas Kanade Pyrmidization → RANSAC

Figure 7: Flow diagram of Optical Flow Tracking.

The Shi-Tomasi corner detector is an improved version of the Harris Corner detector that uses minimum Eigen values instead of the Harris corner matrix[5]. The detected Shi-Tomasi Corner Points are used as the points of interest in the Lucas-Kanade Optical Flow algorithm[6], which tracks objects in video based on the following assumptions:

1. The images are acquired at a sampling rate such that the displacement of the objects in the image due to motion is small.
2. The brightness of a point remains the same even though its spatial location may change.
3. Points move like their neighbors.

Assuming these conditions, the Lucas-Kanade is equation is derived in[6] and the least square solution is shown:

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

Where I_x and I_y are defined as the horizontal and vertical image gradients, u and v are horizontal and vertical velocities, and I_t is the derivative of the image gradient with respect to time. The Lucas-Kanade algorithm is applied to a window of 3 by 3 pixel grid surrounding the Shi-Tomasi point of interest and calculated across multiple Gaussian pyramid scales, yielding optical flow gradient vectors for the input points of interest.

The Lucas-Kanade optical flow pipeline yielded significant false positives despite being able to track the intruder sUAS under certain conditions. The false positives are detected when the intruder blended in with the trees, places in the field, and in the skyline. sUAS detection is confounded by both the SR22 and MigCC traveling at 60 mph, thus the landscape and skyline changes at a high rate with edges, corners, and objects significantly changing their visual

appearance between frames. The conditions when successful tracking occurred when the scale SR22 was moving across the blue sky without busy background variation. When the Lucas-Kanade method is applied to the scale SR22 moving over the trees, the keypoint from the spec is lost quickly due to confusion of the keypoints detected in the image texture of the trees and the lack of robustness from the keypoint. Another contributing factor to keypoint confusion is 3D rotation of the intruder relative to the MigCC. The 2D image of the intruder changes as the perspective of the MigCC relative to the intruder changes and consequently the single keypoint is lost.
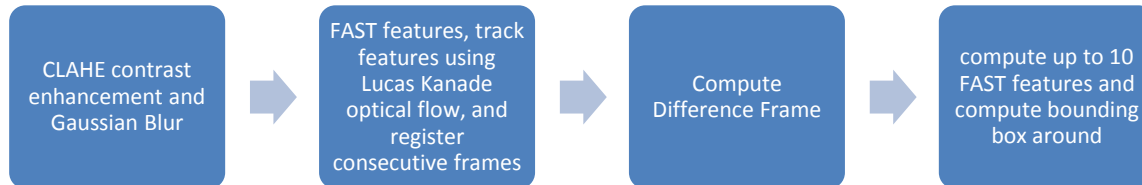
C. FAST Feature

Figure 8: FAST Feature Pipeline

This pipeline includes several steps that are shown in Figure 8 and are now described in detail. The grayscale image is enhanced using the Contrast Limited Adaptive Histogram Equalization (CLAHE), which improves contrast in localized regions of the image. A Gaussian smoothing kernel is used to prevent an excessive number of features from being extracted in the next step. Image registration between consecutive frames is achieved by tracking FAST corner features using the Lucas-Kanade optical flow algorithm. The FAST intensity difference threshold parameter is adaptively set to limit the number of FAST features extracted in each frame while keeping the strongest keypoints. The Lucas-Kanade algorithm operates in the same way as in section 4b except the inputs are FAST features. The perspective transform is computed with the tracked keypoints using RANSAC and the previous frame is aligned to the current frame using the computed transform matrix. Now that the consecutive frames are aligned, their difference image is computed. Moving objects in the video exhibit displacement in the difference frames. FAST features are now computed in the difference frame and then dilated for drawing bounding boxes as shown in Figure 9 below. For convenience of the viewer, the bounding boxes are overlaid over the cropped unprocessed color frame.

The proposed algorithm effectively detected and tracked the intruder sUAS when it is visible on the clear days as shown in Figure 9. False positives did occur in the sky and also in the texture regions of the woods as shown in Figure 10. The high texture wooded regions and boundaries between trees, fields, and roads presents a challenging problem with the number of generated corner keypoints and the similarity of multiple keypoints in a local area. To mitigate for this problem, a simple horizon detector is used to mask the difference frame and limit detecting to only above the horizon, yielding a more robust result above the horizon as shown in Figure 11. The result is a robust, above the horizon tracker. There are still a few false positives but they tend to be reasonable (tip of a cloud for instance). We postulate that a human pilot would need to double check regions of clouds that have the shape of an sUAS at these distances, thus perhaps some of these false positives should be classified as "potential sUAS" or objects that are worthy of monitoring to determine whether they are intruders. A below the horizon sUAS detection methodology is also being developed with a separate set of parameters and processing steps to handle high texture regions.

There are several methods that are currently being pursued to improve sUAS tracking. First the simple horizon detector draws a best fit straight line across the horizon using the canny edge detector and Hough lines. The simple detector fails sometimes during turns due to the confusion of straight lines from roads and tree-lines, especially when the MigCC achieves a high roll during turns and the distortion associated with the standard 170 lenses causes the horizon to turn into a line with significant curvature. Horizon detection will improve with the new non-distortion lenses planned for the next flight test. An additional elastic fine-tuning adjustment using the strongest canny edges should improve accuracy of sUAS detection by further reducing the number of false positives along the horizon. The next planned experiments will use narrower field of view non-distortion lenses and will decrease the distortion associated with the horizon.

Another improvement will be the use of a multi-frame tracking algorithm that will track keypoints from frame-to-frame. Some false positives exist in a single frame of the video while others are intermittent or steady for a few

frames. The moving sUAS tends to generate keypoints that exist across several frames. Thus sUAS detection will improve through the utilization of temporal information and giving greater emphasis on keypoints that are tracked across several frames. The bounding box around keypoints that are tracked across multiple frames should be a different color to indicate robust tracking.

Finally using lenses with field of views of 47 degrees instead of 170 degrees will increase the number of pixels on the sUAS by approximately 3 times for a given distance (ignoring the effects of fish-eye distortion). This is needed for robust sUAS detection at the distances expected (200m to 2000m) so that more robust geometry is captured in the video. A smaller field of view provides the opportunity of several keypoints on the sUAS in locations such as wing planform or wing tip edges, nose points, vertical and horizontal tails, landing gear and wing-fuselage junction.

Figure 9: Moving vehicle detection. *The sUAS is shown in the green bounding box. Another bounding box on the right side of the image shows a moving truck on a highway.*

Figure 10: False positives below horizon. *Six false positives exist below the horizon.*

Figure 11: Detection limited to above horizon. *To mitigate for false positives from the frame shown in 1, an image mask is used to limit detection to above the horizon. The result is a more robust "above the horizon sUAS detector".*

V. Selected University Partner Results

The university partners approached the sUAS detection problem using diverse methods and tools as shown in Table 1. Two groups (The University of Michigan and University of Cincinnati) opted for Convolution Neural Networks (CNN) and reported a lack of training data needed to train neural networks. A summary description of the most successful model from each of the two University partners that provided a positive detection result is below.

Michigan uses a convolution neural network described in detail in their project report. Their detection accuracy is 75% using BVLC_caffenet using three datasets constructed from UAS in the NAS data: training and validation (7285 + 2000 patches sized 256 by 256 from video containing sUAS) and a test set containing 400 frames having fixed wing aircraft. The patches utilized by Michigan contained three object classes (Bixler, sr22, and y6) and three negative classes (sky, cloud, and debris). The test dataset used was labeled "null" by NASA to indicate that there were not sUAS in the video; however, there was a fixed wing full-sized aircraft in the distance in some frames. The object class included Bixler and Y6, however, NASA has determined that those are in fact fixed wing aircraft or helicopter in the distance. The fixed wing aircraft has a slightly larger appearance compared to sUAS in much of the dataset.

The proposed model is of great interest because future flight tests include a narrower field of view which will give more pixels per sUAS, which may be equivalent to the fixed wing aircraft sightings used in training and testing in the BVLC_caffenet. The datasets and CNN constructed in this work will be useful in the future for sUAS and fixed-aircraft detection.

The University of Texas at Arlington implemented an object detection and tracking method for above the horizon using these steps: 1) identify which parts of the image are in the foreground and which parts are in background 2) subtract background from frame to detect blobs 3) A rule-out process eliminates blobs based on minimum blob size, maximum blob size, and total number of possible blobs allowed 4) apply Kalman filter to predict direction of the sUAS. They use a video segment where sUAS is present and achieve detection rate of ~80% with a false positive rate of 90 per frame. Here the false positive rate is defined as the number of incorrect detections for a given frame.

Table 1: Selected University Partner Approaches.

University Partner	Methods	Result
University of Michigan, Dr. Atkins	Above horizon blob detector and Convolution Neural Network (Caffe). UAS in the NAS video is split into smaller datasets for training and testing sets.	Blob detector: slow and was inaccurate CNN: Request more data for training, reported accuracy is subject to overfitting of the NN.
University of Texas at Arlington, Dr. Dogan	Gaussian Mixture Models, Motion based Kalman Filtering, Optical Flow, and color (all MATLAB)	Varied, but generally high detection rate with high number of false positives (85 to 5000+ false positives for a given frame with sUAS).
University of Cincinnati, Dr. Kumar and Dr. Cohen	Convolution Neural Network (MatConvNet, MATLAB)	No positive detections, however, many insights (without using CNN) on the development of SAA algorithms and synthetic data. Request more training and testing data.

VI. Recommendations

Object detection requires distinguishable features. To improve sUAS detection with more distinguishable features, we suggest developing a system that mimics the human visual system. For periphery vision, wide angle lenses such as 120 degrees are useful. However, the human eye has the ability to detect motion in the peripheral vision, which is not replicated in the current detection systems. The FAA requirement for Snellen acuity of commercial and passenger pilots is 20/20 vision, which corresponds to 60 pixels per degree (ppd) of resolution or 1 arc minute[7]. Thus, SAA systems should meet or exceed 60 ppd to match human pilots. The 120 degree field of view lenses yield an average horizontal 22.5 ppd and 12.7 ppd. For the high acuity of human vision, a high resolution camera with a narrow field of view on a gimbal should be used. The high acuity vision serves to 1) scan expected flight path with pan-tilting as needed 2) provide enhanced information on objects detected in the periphery vision by zooming in with the narrower field of view camera.

We plan to repeat the experiment by swapping the 120 degree field of view lenses with 60 and 47 degree non-distortion lenses (64.0 by 56.8 ppd and 81.7 by 58.4 ppd respectively). The decreased field of view will 1) approach human pilot Snellen acuity 2) provide more robust tracking in the best case scenarios (sUAS on clear background) 3) provide more robust and resilient keypoints when the sUAS is on textured backgrounds 4) decrease the potential of the white sUAS blending in with background on cloudy days.

Further, all sUAS should have cameras for data collection to increase the amount available data to train machine learning algorithms on for sUAS detection. This is a quick way of doubling the usable data with more perspectives of the flight paths. For example, the SR22 may be separated with an altitude difference of 100 meters above the MigCC. In this way, one sUAS will have video with the intruder below while the other will have video with intruder above. The extra data will increase the number of types of scenarios and also provide the much needed training data that for CNN and other classifiers.

VII.   Conclusion

NASA and university partners explored several sUAS detection models and present several working algorithms with varying degrees of success. The detectors in this work still need improvement to achieve the goal of operating sUAS beyond line-of-sight; however, much was learned about the current performance and limitations of sUAS detection for these distances and hardware. Further, several areas of improvement were identified. The next flight test will use non-distortion lenses with smaller field of view to mitigate for keypoint dropout over texture backgrounds by providing more pixels per sUAS. Future flight test opportunities are essential to further development of the integration of sUAS in the NAS.

I.   References

[1] Prats, X., Delgado, L., Ramirez, J., Royo, P., Pastor, E., "Requirements, Issues, and Challenges for Sense and Avoid in Unmanned Aircraft Systems," *Journal of Aircraft*, Vol. 49, May-June 2012.

[2] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "SURF: Speeded Up Robust Features ," *Computer Vision and Image Understanding*, Vol. 110, 2008, pp. 346-359.

[3] Bradski, G. and Kaehler, A., "Learning OpenCV: Computer vision with the OpenCV Library", [software library], 2008.

[4] Lowe, D., "Distinctive image features from scale-invariant keypoints," *International Journal Computer Vision*, Vol. 60, No. 2, 2004, pp. 91-110.

[5] Harris, C., and Stephens, M., "A Combined Corner and Edge Detector," *Proceedings. Alvey Vision Conference,* 1988, pp. 147-151.

[6] Lucas, B. and Kanade, T., "An iterative image registration technique with an application to stereo vision," *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981, pp. 674– 679.

[7] Bailey, R., Wilz, S., Arthur, T., III, T., "Conceptual Design Standards for eXternal Visibility System (XVS) Sensor and Display Resolution," *NASA Technical Memorandum 2012-217340*, 2012