Development of a Peripheral–Central Vision System for Small Unmanned Aircraft Tracking

Changkoo Kang,[*] Haseeb Chaudhry,[†] Craig A. Woolsey,[‡] and Kevin Kochersberger[§]
Virginia Tech, Blacksburg, Virginia 24061

https://doi.org/10.2514/1.I010909

Two image-based sensing methods are merged to mimic human vision in support of airborne detect-and-avoid and counter–unmanned aircraft systems applications. In the proposed sensing system architecture, a peripheral vision camera (with a fisheye lens) provides a large field of view, whereas a central vision camera (with a perspective lens) provides high-resolution imagery of a specific target. Beyond the complementary ability of the two cameras and supporting algorithms to enable passive detection and classification, the pair forms a heterogeneous stereo vision system that can support range resolution. The paper describes development and testing of a novel peripheral–central vision system to detect, localize, and classify an airborne threat. The system was used to generate a dataset for various types of mock threats in order to experimentally validate parametric analysis of the threat localization error. A system performance analysis based on Monte Carlo simulations is also described, providing further insight concerning the effect of system parameters on threat localization accuracy.

Nomenclature

C^c	=	field-of-view coverage of the central vision camera
f^c	=	focal length of the central vision camera
f^{c*}	=	optimal focal length of the central vision camera
f_h^{c*}	=	optimal focal length of the central vision camera in the horizontal direction
f_v^{c*}	=	optimal focal length of the central vision camera in the vertical direction
f^p	=	focal length of the peripheral vision camera
H_s^c	=	sensor height of the central vision camera
H_t	=	threat height
$R_{c/p}$	=	rotation matrix between the peripheral and central vision frame
r	=	threat range
r_a	=	assumed threat range
$\boldsymbol{r}_{p/c}$	=	optical center position of the peripheral vision camera in the central vision frame
$\boldsymbol{r}_{p/g}$	=	optical center position of the peripheral vision camera in the global reference frame
$\boldsymbol{r}_{t/c}$	=	threat vector from the central vision camera
$\boldsymbol{r}_{t/p}$	=	threat vector from the peripheral vision camera
$\boldsymbol{r}_{t/g}$	=	estimated threat position in the global reference frame
$T_{c/p}$	=	translation vector between the peripheral and central vision frame
u^p, v^p	=	pixel coordinates of the threat on the peripheral vision image
v_h	=	relative horizontal speed of the threat in the central vision frame
W_s^c	=	sensor width of the central vision camera
W_t	=	threat width
δ_r	=	localization error
δ_s	=	sensor error
θ	=	intersection angle between threat vectors

I. Introduction

WITH the rapid proliferation of small unmanned aircraft systems (sUAS), the risk of midair collisions is growing, as is the risk associated with the malicious use of these systems. The airborne detect-and-avoid (ABDAA) problem and the counter-UAS problem have similar sensing requirements for detecting and tracking airborne threats, although for different purposes: to avoid a collision or to neutralize a threat, respectively. These systems typically include a variety of sensors, such as electro-optical or infrared (EO/IR) cameras, RADAR, or LiDAR, and they fuse the data from these sensors to detect and track a given threat and to predict its trajectory. Camera imagery can be an effective method for detection as well as for pose estimation and threat classification, though one cannot resolve range to a threat from a single camera image without additional information, such as knowledge of the threat geometry.

Although a variety of sensors are available, including those mentioned above, cameras are the most common sensor for sUAS. As the size, weight, and power and cost (SWaP-C) of cameras has continued to drop, whereas resolution and image quality have continued to improve, and as computer vision methods have continued to develop, more information can be extracted from camera imagery than was possible in the past [1]. Considering their low SWaP-C and the expanding capabilities of computer vision, cameras will likely play an important future role in allowing sUAS to detect and track other aircraft that may pose a navigational hazard or a malicious threat.

In this paper, a peripheral–central vision (PCV) system that detects, localizes, classifies, and tracks aircraft using only two low-cost cameras is introduced. The focus here is on detecting and characterizing sUAS at close range (tens to hundreds of meters), but the concepts and algorithms can be extended to other applications involving larger aircraft using more capable camera systems. Fast and reliable initial threat detection is crucial for an ABDAA or C-UAS system, so the camera system must be able to see a large area at once. This observation suggests the use of an omnidirectional peripheral vision camera for the initial threat detection. To provide continuous visual coverage of the environment for threat detection, however, requires a wide field-of-view (FOV) camera, the peripheral vision sensor. To classify a threat aircraft, estimate its pose, and better predict its flight path, a higher-resolution image is required, which suggests the use of a gimbal-mounted perspective camera, the central vision sensor, with a narrower FOV. Incorporating each type of sensor affords the opportunity to use stereo vision for ranging. Accordingly, this paper introduces a heterogeneous PCV system for use in ABDAA applications or in a ground- or air-based Counter-Unmanned Aircraft Systems (C-UAS) system. The PCV system is capable of detecting flying objects within a wide FOV, classifying these threats, and estimating their position (including range), attitude, and velocity.

Received 29 September 2020; revision received 20 February 2021; accepted for publication 21 February 2021; published online 19 April 2021. Copyright © 2021 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 2327-3097 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

[*] Graduate Research Assistant, Department of Aerospace and Ocean Engineering.

[†] Graduate Research Assistant, Department of Mechanical Engineering.

[‡] Professor, Department of Aerospace and Ocean Engineering. Associate Fellow AIAA.

[§] Associate Professor, Department of Mechanical Engineering. Associate Fellow AIAA.

The main contribution of this paper is analysis and testing of a novel air-based PCV system that detects, locates, and tracks airborne threats. The paper is an extension of work presented in [2]. The peripheral vision camera first detects a threat and cues the central vision camera based on the viewing angle. The central vision camera then slews into position to focus on the threat, enabling classification and ranging. This paper describes the development and testing of the PCV system architecture and algorithms. Section II describes prior work related to the PCV system. Section III describes the system setup. Image processing methods for the peripheral vision imagery, including threat detection algorithms, are described in Sec. IV, and the cueing algorithm that directs the central vision camera is presented in Sec. V. Image processing methods for the central vision imagery are described in Sec. VI. Section VII describes a heterogeneous stereo vision algorithm for threat localization. Section VIII describes the system architecture and data management. A more general discussion of system performance is provided in Sec. IX. Section X describes analysis of threat localization performance using experimental data. Section XI presents conclusions and summarizes ongoing work.

II. Related Work

Because passive sensors do not radiate energy, they require less power than active sensors, making them attractive for use on weight-constrained sUAS. Dramatic advances in camera hardware and image processing software make machine vision systems especially appealing, both for mission-related tasks such as aerial imagery and for operational tasks like ABDAA. The need to localize threats once they are detected, however, suggests the use of multiple cameras to allow for stereo vision, which enables depth estimation and 3D reconstruction. These systems typically include two identical perspective cameras, but the narrow FOV of a perspective camera limits its utility for sUAS detection. To address this issue, Drulea et al. [3] and Kita and Kita [4] proposed using a stereo fisheye vision system to relax the FOV limitation. However, fisheye cameras provide lower pixel coverage than a narrow FOV perspective camera, in a given region, making it more difficult to classify a detected threat or to estimate its position and attitude.

Earlier efforts have combined the advantages of a large FOV (“omnidirectional”) camera and a narrow FOV perspective camera for surveillance, focusing mainly on detecting people [5–8] and on tracking them using facial recognition [9,10]. Baris and Bastanlar [11] used such a dual-camera system to classify objects in a scene and to improve surveillance performance. The dual-camera systems used in these studies showed good performance for tracking targets from a fixed location. The omnidirectional camera, however, was used primarily for initial detection and to cue the perspective camera. Because the focus of these studies was surveillance, there was no effort to extract stereo image data (e.g., range to a threat) from the camera pair. However, Muñoz-Salinas et al. [12] suggested an algorithm based on particle filtering to localize people in a scene using multiple heterogeneous cameras. The proposed algorithm provides estimated locations of people, along with confidence data obtained from the particle filter. For a different application other than surveillance, Eynard et al. [13–15] suggested an algorithm to estimate the altitude and motion of an unmanned aircraft using an onboard, heterogeneous stereo vision system that consists of a fisheye lens camera and a perspective camera. Their algorithm first finds the *homography matrix* [16–18] that relates the two camera views and then estimates the distance between the horizontal plane (i.e., the tangent plane to the Earth’s surface) and the first camera. The algorithm determines the altitude of the “own ship” aircraft, but does not provide information about other aircraft in the FOV.

Most applications of heterogeneous camera systems involve surveillance of confined areas (<20 m range) from a fixed location. Few studies discuss stereo ranging using a heterogeneous camera system for the ABDAA and C-UAS applications. The following sections describe a PCV system that detects, localizes, classifies, and tracks the motion of sUAS.

III. PCV System Setup

Referring to the process diagram for the proposed PCV system in Fig. 1, threat detection and characterization begins at the left with a detection by the peripheral vision system. First, the lens distortion of the peripheral vision imagery is removed. The undistorted imagery is then stabilized using a homography matrix [16–18] computed from the image sequence. Threats are then detected by their motion through the undistorted, stabilized imagery using optical flow. The bearing angle to each detected threat, measured relative to the boresight axis of the peripheral vision camera, is then converted to an approximate bearing angle in a frame fixed in the central vision camera. In scenes with multiple detected threats, the system selects one according to a scheduling algorithm (see [19–22], for example) and provides the bearing angle as a cue for the gimbal that controls the central vision camera’s attitude. The central vision camera slews to the indicated direction and the threat is then detected in the central vision image. The higher-resolution image from the perspective camera enables the classification of the threat using a deep neural network (DNN) and the estimation of its attitude using the visual pose estimation process presented in [1]. In addition, a heterogeneous stereo vision strategy uses the two distinct “threat vectors” obtained from the two camera images to estimate the threat position.

For the proof-of-concept analysis and testing, we built and developed a system with two camera sensors as shown in Fig. 2. The Insta360 Air was selected as the peripheral vision camera because this camera contains two fisheye lenses, providing 360° coverage in both azimuth and elevation, and is compatible with a variety of embedded hardware systems and with widely used software tools such as those available through OpenCV and the robot operating system (ROS). The GigE DFK Z12G445 color zoom camera from The Imaging Source serves as the central vision camera. This global shutter camera captures images at up to 41 frames per second (fps) and has a software-controllable, motorized 12× optical zoom lens. The GigE camera is mounted on an HDAir Infinity MR S2 gimbal, which enables the camera to be directed by the cue provided from the peripheral vision

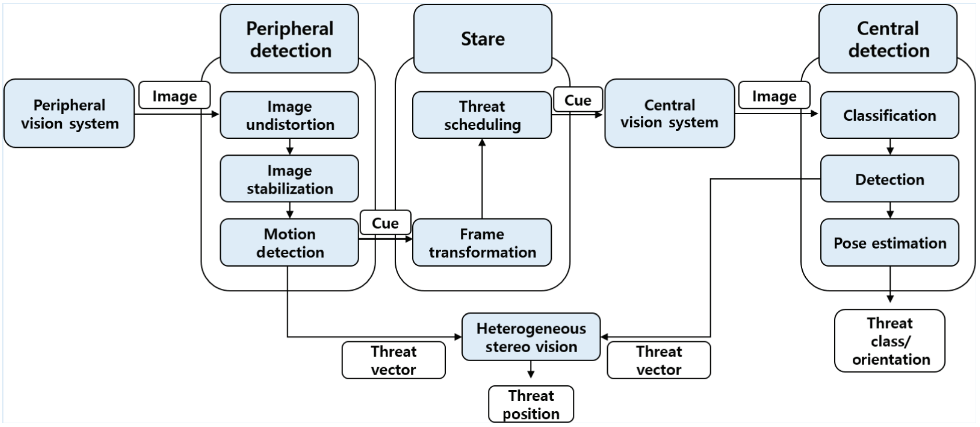


Fig. 1 The flowchart of the PCV system.

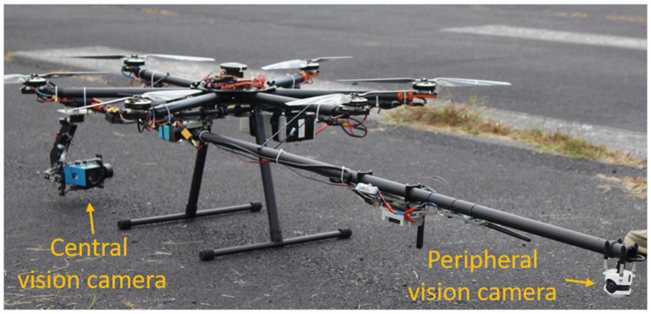


Fig. 2 Peripheral–central vision system setup.

camera. These two cameras are accessed using ROS, installed on an Nvidia TX2. This on-board computer runs all system algorithms, such as detection and cueing. The detailed camera specifications are shown in Table 1. Figure 2 shows the PCV system setup, as configured for air-based operation. The data stream from the air-based system is sent to the ground station wirelessly. For ground-based testing, however, the PCV system was connected directly to the ground station by cable. Data collection using these two PCV system setups is described in Sec. VIII.

IV. Peripheral Vision Subsystem

Although the peripheral vision camera system provides complete coverage, the images have relatively low resolution and high distortion. The number of pixels associated with an object at a given range is low, compared with a perspective camera, and varies nonlinearly over the camera FOV. As an example, for the Insta 360 Air, which is our peripheral vision camera, the pixel width of a 5 m object at 100 m distance is only 22 pixels, as shown in Fig. 3a. When there are enough pixels on a threat, the peripheral camera can provide information that is useful in characterizing the threat, but the lens introduces a high level of image distortion, especially at the edges of the image. Distortion can be partially addressed through proper camera calibration [23], but the pixel density is unavoidably lower away from the camera boresight. As an example, Figure 3b shows the relative pixel density for an undistorted Insta360 Air image versus angle from boresight.

Table 1 Camera specifications

Parameter	Peripheral vision camera (Insta360 Air)	Central vision camera (GigE DFK Z12G445)
Focal length, mm	1.0	4.8–57.6
Sensor size, mm	3.3 × 3.3	4.8 × 3.6
Pixel size, μm	2.19 × 2.19	3.75 × 3.75
Resolution, px	1504 × 1504	1280 × 960
Size, mm	ϕ36.6 × 39.6	50 × 50 × 103
Weight, g	26.5	330

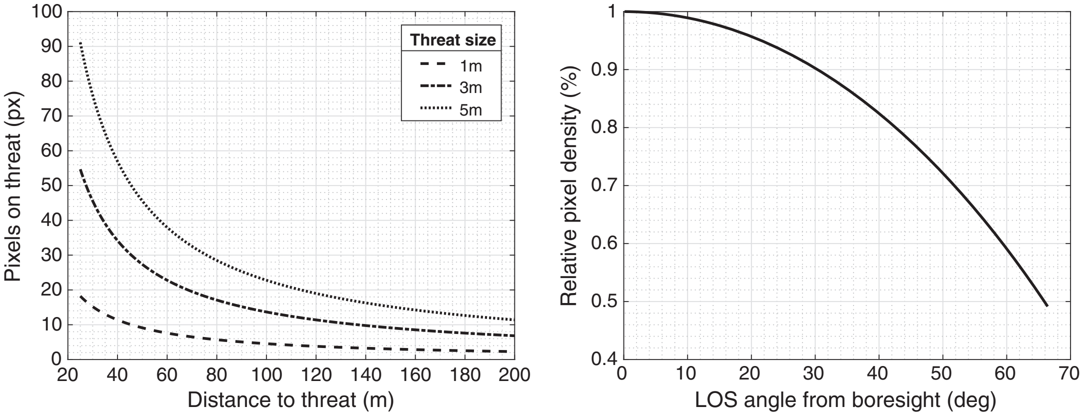
Image resolution is less of a concern in threat detection than in the other image processing tasks. This is because detection methods such as the optical flow algorithm used here [24,25] can detect moving objects that occupy a small number of pixels. A peripheral vision camera system can be quite useful for initial threat detection because of its large FOV. Having detected a threat, a central vision camera may be cued to investigate further. If additional information is available from the peripheral vision image, it can be combined with central vision imagery to improve overall situation awareness. The complementary nature of the peripheral vision camera's wide FOV and the steerable central vision camera's high resolution motivated the proposed architecture.

A. Image Preprocessing

For a camera that is fixed in space, the pixels associated with static objects and with the unmoving background do not change state between images. For an airborne system, however, the camera translates and rotates as the aircraft moves, so the background and static objects appear to move within the image sequence. This apparent motion of static elements due to motion of the camera must be eliminated in order to detect moving objects using optical flow.

A homography matrix indicates the relative rotation and translation between two images of a given scene [16–18]. The homography matrix between consecutive frames of a moving camera can therefore be used to remove the apparent motion in the image. This process is called image stabilization in the computer vision community; it is a software variant of mechanical image stabilization. Computing a homography matrix requires the pixel locations of common feature points in the sequence of images. A Harris corner detector [26] is applied to the peripheral camera imagery to find feature points. The optical flow algorithm is then used to track detected feature points in the sequence of images. From the sequence of pixel locations of selected feature points, the homography matrix is estimated with the RANSAC algorithm [27], which generates an optimal estimate while excluding outlier feature points. The homography matrix is estimated for each new image frame and is then used to remove the effect of camera motion by rotating and translating the image.

Homography-based image stabilization performs quite well in nominal conditions, but the algorithm can be affected by lighting conditions and image noise. As an alternative, one may consider image stabilization based on direct measurements of camera motion, obtained using an inertial measurement unit (IMU). This approach compensates for camera motion using camera pose data from the IMU, rather than homography data extracted from images. This IMU-based approach requires that the image and inertial motion data be accurately synchronized for good performance, however, and the IMU's accuracy is an important performance factor. Moreover, the IMU-based approach cannot accurately account for the camera's translational motion, which



a) Pixels on threat versus distances from camera

b) Relative pixel density of the undistorted image

Fig. 3 Capabilities of Insta360 air.

is an effect that is included implicitly in the homography-based approach. The example processed image is shown in Fig. 4.

To leverage the strengths of both approaches, a data fusion Kalman filter [28] was developed. The fused camera motion data are converted to a rotation matrix that is used to stabilize images for use in optical flow-based threat detection. The flow chart in Figure 5 illustrates the image stabilization process.

B. Threat Detection Using Optical Flow

Various computer vision algorithms have been proposed to detect a moving threat using visual imagery. Attributes that are unique to a particular scenario can pose special challenges or opportunities for visual detection. For a C-UAS system, for example, flying objects may appear with strong contrast against a static background (e.g., a clear blue or overcast sky). Non-antagonistic aircraft, including many sUAS, may even include lighting to make them more visible.

In the approach described here, an optical flow algorithm computes the translational displacement of pixels in consecutive images. Because the pixel coordinates change for an object moving through an image against a static background, such pixel displacements may indicate threats. To detect a threat, several feature points within the image are extracted using a corner detector. The optical flow algorithm is then applied to track these feature points in consecutive images. Pixel velocity vectors whose magnitude exceeds a threshold indicate candidate threats. Setting a high threshold may result in missed detections, whereas a low threshold value may create false detections. To explore the sensitivity of threat detection to conditions, we generated representative receiver operating characteristic (ROC) curves for optical flow detection. Two experiments were performed. In experiment 1, the threat was located relatively close (20–30 m) to the PCV system. In experiment 2, the threat was farther away (50–60 m). Figure 6 shows an ROC curve of each of these two experiments with detection threshold values varying from 0 to 5.5 px. Note the "knees" in the two ROC curves that indicate appropriate threshold values for detection. For the closer threat, the knee occurs at a threshold value of 3.5 px. For the more distant threat, the indicated threshold was 1.5 px. Threshold values were scheduled based on range to the threat. For initial detections at unknown range, a low threshold value (corresponding to the maximum detectable range) is used. Because we assume that there is at most one threat, and that this threat is moving against a static background, the region of the image with the fastest pixel speed is declared to be a candidate threat for further investigation. After confirming that the candidate is indeed a threat and estimating its range, using an algorithm described in the following sections, the threshold value is adjusted to correspond to the estimated threat range.

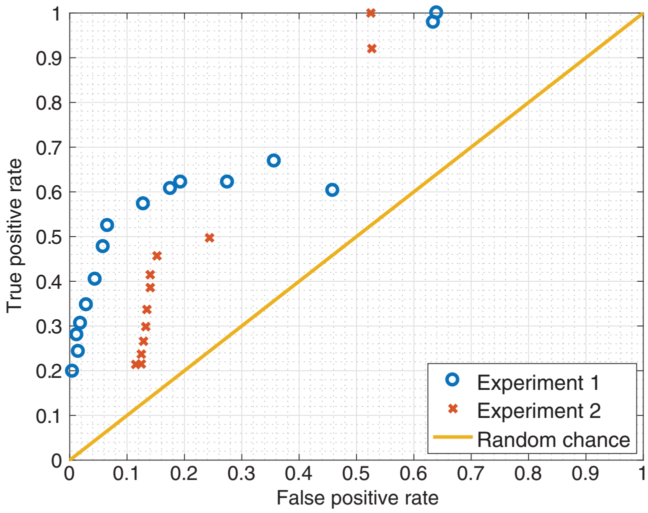


Fig. 6 Receiver operating characteristic (ROC) curve of optical flow detection.

Special challenges arise in vision-based threat detection as proposed, and these are the subject of continuing study. For example, a threat coming straight toward the camera may not be detected because the optical flow algorithm detects relative motion in the image frame. Also, a dynamic or variegated background will increase

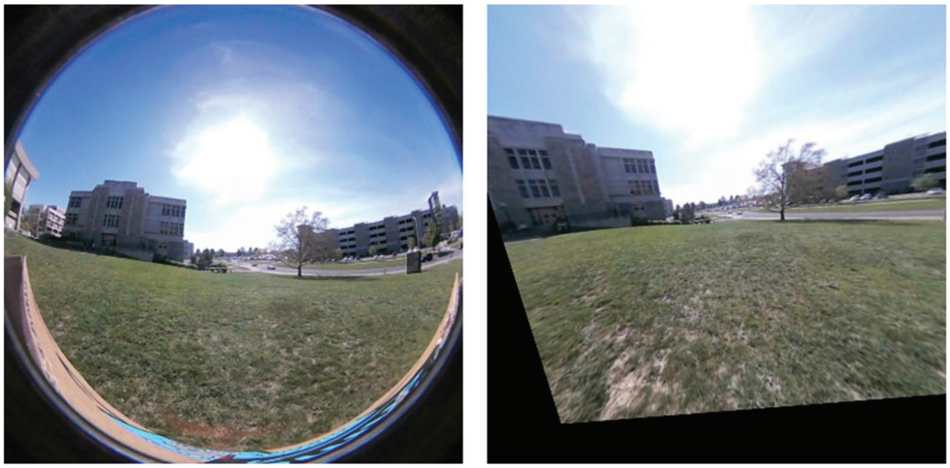


Fig. 4 Peripheral vision image before (left) and after (right) undistortion and stabilization.

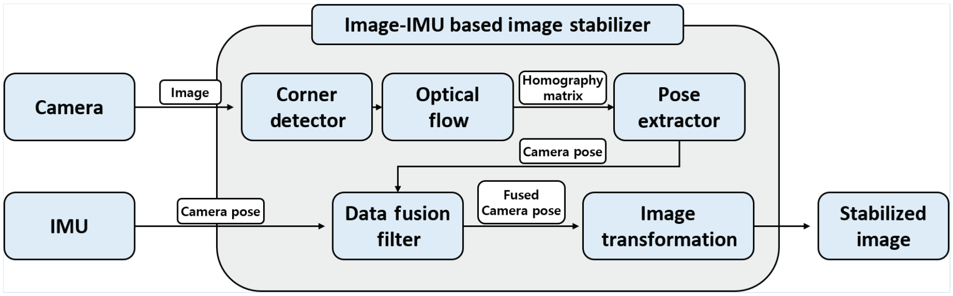


Fig. 5 Homography–IMU data fusion for image stabilization.

the number of false detections. This problem may be addressed in part by adaptively tuning the detection threshold values and by incorporating complementary feature-based detection algorithms such as Scale-Invariant Feature Transform (SIFT) [29], Speeded Up Robust Features (SURF) [30], or Oriented FAST (Features from Accelerated Segment Test) and Rotated BRIEF (Binary Robust Independent Elementary Features) (ORB) [31] with tracker algorithms such as Kernelized Correlation Filter (KCF) [32] or Multiple Instance Learning (MIL) [33].

After a threat is detected in the peripheral camera imagery, a Kalman filter estimates its bearing in the central vision camera frame and cues the gimbal toward the threat. For multiple threat detections, a scheduling algorithm can provide an optimal cueing sequence [19]. When the optical flow algorithm loses track of a threat, the Kalman filter predicts the threat's pixel location based on its last known pixel velocity. If and when a direct measurement of the threat location becomes available once again, the Kalman filter corrects the threat location estimate that serves as a cue to the central vision camera. Figure 7 shows example results for two (undistorted) images obtained using the peripheral vision camera with Kalman filtering. The red circle on the left indicates a moving object detected by optical flow. The Kalman filter corrects its position estimate (green dot) based on the detection. There is no optical flow detection on the right; the Kalman filter predicts the threat position based on the last estimated pixel position and velocity. Optical flow appears to be effective at detecting candidate threats against a static background, even for threats of small pixel size. One may use the threat location and velocity in pixel coordinates, as described in the next subsections, to cue the central vision camera system, which can then classify the threat and estimate its position and attitude.

V. Cueing Algorithm

The threat detected by the peripheral vision camera must then be observed by the central vision camera to obtain more detailed information such as position, velocity, attitude, and classification. For the gimbaled central vision camera to slew to the threat, a cue is required from the peripheral vision system. This section describes a cueing algorithm that computes the azimuth and elevation angle of the threat in the central vision camera-fixed reference frame. The section also discusses the error in this threat cue and its effect on threat acquisition by the central vision camera system.

A. Threat Bearing Angle

Having detected a threat and extracted its pixel coordinates from a peripheral vision image, one may estimate the azimuth and elevation angles to the threat in a reference frame fixed in the peripheral vision camera. These angles are then transformed to a frame fixed in the central vision camera. Given the relative pose between the peripheral ("p") and central ("c") vision cameras, as defined by the proper rotation matrix $R_{c/p}$ and the translation vector $T_{c/p}$, the threat vector in the central vision camera-fixed reference frame is

$$\boldsymbol{r}_{t/c} = \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = \begin{bmatrix} R_{c/p} | T_{c/p} \end{bmatrix} \begin{bmatrix} \dfrac{u^p r_a}{f^p} \\ \dfrac{v^p r_a}{f^p} \\ r_a \\ 1 \end{bmatrix} \tag{1}$$

where u^p and v^p are the pixel coordinates of the threat in the peripheral vision image, f^p is the focal length of the peripheral vision camera, and r_a is an assumed range to the threat. The actual range is unknown, at this point, because the threat has only been imaged by the single, peripheral vision camera. However, a range to the threat is needed in order to compute the bearing angle in the central vision camera-fixed reference frame. There are some methods for estimating range using a monocular camera, such as depth-from-focus/defocus [34–36]. Because these methods are not robust for distant objects, however, we instead adopt an *assumed* range to obtain the initial cue for the central vision camera and then replace this assumed range with a more accurate estimate once the threat has been acquired by both cameras.

B. Optimal Assumed Range

Because an assumed range is used, the error between actual range and the assumed range may lead the central vision camera to miss the object to which it has been cued. For example, if the assumed range is 30 m, but the actual range is 90 m, the azimuth angle error would be −3.5°. If the actual range is 10 m, then the azimuth angle error would be 9.7°. If this cueing error is large, the threat may lie outside the central vision camera's FOV and fail to be acquired. To resolve this issue, an assumed range value that minimizes the cueing error should be chosen.

Figure 8a shows the azimuth error versus actual range to a threat for four assumed ranges varying from 10 to 100 m. (The maximum range at which the given peripheral vision system can detect a sUAS is roughly 100 m.) When the assumed range is 10 m (blue dotted curve) or 100 m (purple curve), the maximum absolute bearing error is nearly 15°. On the other hand, when the assumed range is 30 m (red dashed curve), the maximum absolute bearing error is 10°, and less than 4° for threats beyond 20 m. An optimal assumed range that minimizes the maximum absolute bearing angle error is indicated in Fig. 8b, which shows the maximum absolute bearing angle error versus assumed range. As shown in this figure, when the assumed range is 17.93 m, the maximum absolute azimuth angle error is minimum, at roughly 7.3°. The maximum elevation angle error is much smaller than the maximum azimuth error and it is relatively insensitive to the assumed

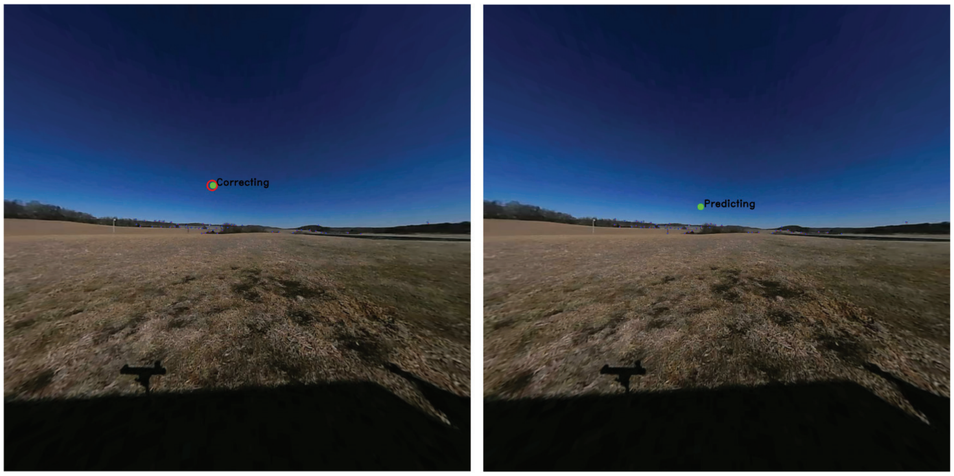
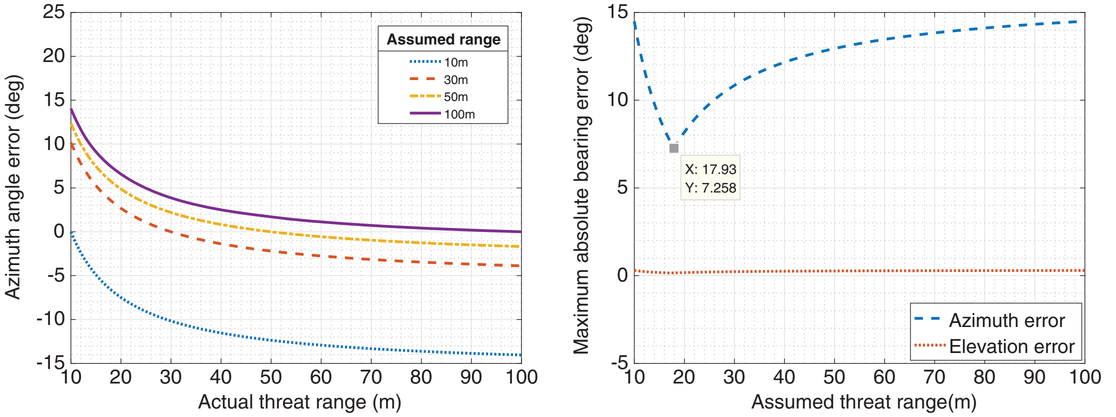


Fig. 7 Two image frames illustrating the results of Kalman-filter-aided optical flow detection.



a) Bearing error for four assumed ranges

b) Maximum absolute bearing error versus assumed range

Fig. 8 Optimal assumed range estimation.

threat range, although larger elevation angle errors may result for threats that are high above the horizon. In any case, an assumed range of 17.93 m is used for the given system.

VI. Central Vision Subsystem

Although the peripheral vision camera system has a large FOV and low resolution, the pan-tilt-zoom (PTZ) central vision camera has a narrower FOV, but high resolution. The central vision camera is intended to obtain detailed imagery of a threat that has been detected by the lower resolution peripheral vision camera. Once the central vision camera has slewed toward the cue provided by the peripheral vision system and acquired the threat within an image, the central vision camera begins to track and classify the threat and to estimate its position, attitude, and velocity.

A. Gimbal Control for Pan and Tilt

The reference azimuth and elevation angles computed from peripheral vision imagery are sent to the gimbal, which uses a PID controller to rotate the camera to the desired attitude. The gimbal controller uses Kalman filtered orientation measurements that incorporate data from an IMU fixed to the rotating camera frame and from the gimbal's servo axis encoders. Because the gimbal's IMU lacks a magnetometer, the gimbal's yaw axis encoder angle is used as the primary azimuthal orientation reference. The readings from the gimbal's IMU are recorded synchronously. The quality of the IMU filtered measurements should be considered in determining the system's ranging accuracy. The gimbal manufacturer claims an angular precision of 0.02° in all axes, which has implications for using the central camera's zoom capability; a narrower FOV (e.g., fully zoomed) requires a smaller error in gimbal orientation to maintain the view of a threat during tracking. For the current system setting, the minimum horizontal FOV of the central vision camera is about ±2.5°. The gimbal precision is fine enough to support tracking at full zoom, though accurate tracking also depends on the disturbance environment and the servo-controller performance. In the implementation described here, the central vision camera adopts its widest FOV when tracking the reference cue to increase the likelihood of acquiring the threat.

B. Threat Classification

Object classification can reveal whether a detected object poses a threat and can aid in motion prediction by indicating the performance capabilities of a given threat. A classification taxonomy for detected airborne objects might include coarse categories of aircraft, birds, kites, or floating debris. Finer subclasses for aircraft such as fixed-wing, helicopter, or multirotor can be used with better sensor resolution and algorithmic capabilities. Even finer classifications might include specific models of aircraft. Here, we focus on binary classification as "aircraft" or "not aircraft" for a proof-of-concept.

Airborne threat classification has been of interest in defense scenarios since aircraft began playing a major role in warfare. Personnel were trained, for example, to classify aircraft on the basis of engine sound, visual cross section, or trajectory to determine whether a particular aircraft posed a threat [37]. A wide variety of criteria, such as those mentioned above, can be used in classification algorithms, but a lack of diverse data can impede algorithm development and validation, especially if the aim is to create a rich taxonomy with many subclasses. A broader classification is more feasible, especially for unmanned aircraft that exhibit a wide variety of shapes and configurations. Analytical methods that use feature extraction can discriminate between general categories based on feature presence. For example, propeller count or cross-sectional aspect ratio can inform aircraft class likelihoods based on the type of craft. Simply discriminating between fixed-wing and multirotor sUAS could significantly reduce uncertainty by enabling a tracking algorithm to incorporate an appropriate motion model into the prediction method. Developing a finer classification, using analytical or data-driven approaches, would require proportionally larger and richer data sets. Common strategies to address the limitations of small datasets, such as retraining the last layer for an object detection neural network, become less effective with finer classification schemes. Although simulated datasets can help compensate for a lack of training data, simulation fidelity can play a significant role in the outcome, a topic of ongoing research [38]. Although a refined classification scheme was not our focus in developing the system described here, threat type classification can improve tracking performance by supporting threat model selection.

Several approaches to classification are described in the literature; the ones explored here employ machine learning and computer vision frameworks. Machine learning methods, specifically those involving neural networks (NNs), have been extensively developed for problems involving the detection and classification of objects in images. Existing deep NNs such as MobileNet [39], YOLOv2 [40], and YOLOv3 [41] have been shown to exhibit high "true positive" rates when trained on sufficiently large and diverse datasets. These networks can be accelerated to run in real-time using GPU resources available on flight-capable computational hardware; however, performance depends strongly on the datasets used to train the networks. As mentioned above, last layer retraining can compensate for smaller datasets. In this approach, an existing network's weights are used for feature extraction and only the last layer, where classification occurs, is adapted to a particular use case.

To demonstrate the concept, an implementation of YOLOv3 that was trained on the Common Objects in Context (COCO) [42] dataset was used to generate bounding boxes around aircraft in images obtained using the central vision camera. Some examples are shown in Fig. 9.

Even with a generically trained NN, the aircraft is correctly detected at closer ranges. In edge cases, however, such as when the aircraft descends below the horizon or appears against a less distinct background, the NN tends to fail. Cases with image noise and lower pixel coverage also result in false negatives. Because the focus here is on proof-of-concept rather than performance optimization, the

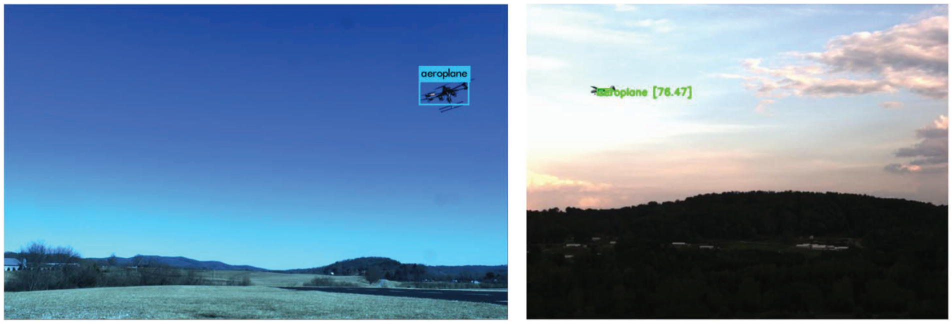


Fig. 9 Example results using COCO-trained YOLOv3.

YOLOv3 implementation used pretrained NN weights. Retraining with a dataset that contains a variety of small UAS operating in a rich set of scenes and environmental conditions would likely yield more robust NN detection and classification.

C. Zoom Optimization

To make use of the central vision camera's zoom capability, a control strategy was developed to optimize the camera's FOV when acquiring and tracking a threat. A low zoom value (large FOV) reduces the chance that a disturbance to the camera's orientation would cause the threat to be lost from view. On the other hand, a high zoom value (narrow FOV) enables better threat detection and characterization by providing more pixels on the threat. Here, we consider how the zoom setting can be adjusted to optimize the tradeoff between these two concerns.

Given the range r to the threat, which can be estimated using the algorithm described in Sec. VII, the FOV coverage C^c of the central vision camera is

$$C^c = \frac{rW_s^c}{f^c} \tag{2}$$

where W_s^c is the sensor width and f^c is the focal length; see Fig. 10. To prevent the threat from leaving the FOV, the speed of the threat relative to the central vision FOV and the margins around the threat within the FOV should be considered. For example, if the FOV coverage of the central vision camera is 10 m at the threat range, and the threat width is 2 m, the margin of the FOV is 4 m. In this case, if the threat relative speed is more than 4 m/s, then the threat will escape the static camera's FOV in 1 s. Therefore, the FOV margin should be larger than the (pixel) distance moved by the threat within a given time period γ_t (say, 2 s), which depends on the system latency as well as the error in the reference cue and in the measurements (e.g., gimbal orientation).

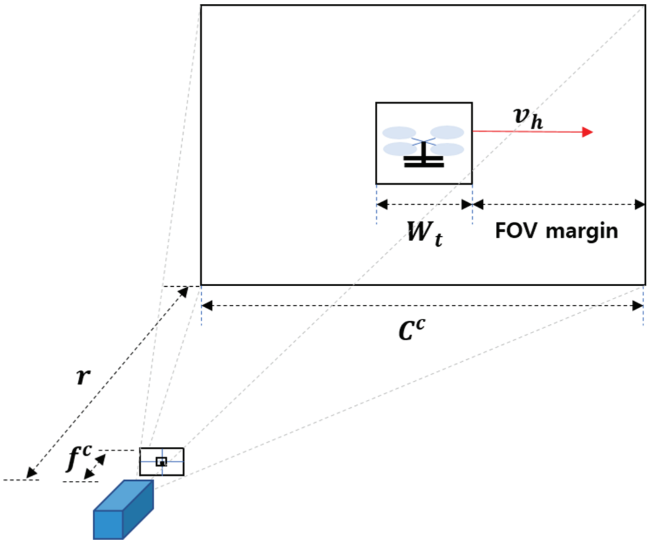


Fig. 10 Zoom optimization geometry.

To formulate an optimization problem, consider that the threat size in the image should be maximized to increase the number of pixels on the threat, but the threat should not escape the camera's FOV in time γ_t. These objectives are captured in the following:

$$\begin{aligned} &\text{Maximize} \quad \left(\frac{W_t}{C^c}\right)^2 \\ &\text{s.t.} \quad \gamma_t v_h - \frac{C^c - W_t}{2} = 0 \end{aligned} \tag{3}$$

where v_h is the relative horizontal speed of the threat and W_t is the width of the threat. Formulating the Hamiltonian

$$H = \left(\frac{f^c W_t}{rW_s^c}\right)^2 - \lambda\left(\gamma_t v_h - \frac{rW_s^c}{2f^c} + \frac{W_t}{2}\right) \tag{4}$$

we compute

$$\frac{\partial H}{\partial \lambda} = \gamma_t v_h - \frac{rW_s^c}{2f^c} + \frac{W_t}{2} = 0 \tag{5}$$

$$\frac{\partial H}{\partial f^c} = \frac{2f^c W_t^2}{r^2 (W_s^c)^2} - \lambda\left(\frac{rW_s^c}{2(f^c)^2}\right) = 0 \tag{6}$$

Solving Eq. (6) for the zoom setting gives

$$f^c = rW_s^c \sqrt[3]{\frac{\lambda}{4W_t^2}} \tag{7}$$

Substituting into Eq. (5) gives

$$\lambda = \frac{4W_t^2}{(W_t + 2\gamma_t v_h)^3} \tag{8}$$

Finally, we substitute Eq. (8) into Eq. (7) to find the focal length that maximizes the threat size in the FOV while ensuring a sufficient margin to prevent losing the threat:

$$f_h^{c*} = \frac{rW_s^c}{W_t + 2\gamma_t v_h} \tag{9}$$

Note that only the horizontal direction (W_t, W_s^c, and v_h) is considered in this formulation. The optimal focal length for the vertical direction is computed in the same way:

$$f_v^{c*} = \frac{rH_s^c}{H_t + 2\gamma_t v_v} \tag{10}$$

where H_s^c is the sensor height of the central vision camera and H_t is the height of the threat. The optimal focal length is then taken as the minimum value of the focal lengths for the horizontal and vertical direction:

$$f^{c*} = \min(f_h^{c*}, f_v^{c*}) \tag{11}$$

VII. Heterogeneous Stereo Vision

A camera image is generated when points on 3D objects are projected onto the camera image sensor. As shown in Fig. 11, the points A, B, and C are projected onto the same image point of the right image sensor. The actual position of the point represented in the image is therefore unknowable, without additional knowledge about the range to the object. This issue is called "range ambiguity." If we have an additional camera, viewing the same scene from a different perspective, the range ambiguity can be resolved by computing an intersection of two lines of sight from the two cameras. The point B in Fig. 11 is an example. Conventional stereo vision algorithms normally assume two identical cameras with parallel camera boresight axes. This configuration allows one to determine the range to an object using a simple equation. The formulation must be modified for a heterogeneous stereo vision system, however, because of high distortion in the peripheral vision image and the nonparallel camera boresight axes. An omnidirectional camera calibration procedure published by Scaramuzza et al. [23] addresses this issue, enabling conversion from 2D image points on the large FOV camera image to corresponding 3D vectors. The 3D vector pointing toward the threat from a given camera is referred to as a *threat vector*. The 3D position of a threat can be estimated by computing the intersection of the threat vectors from the two cameras.

Figure 12 depicts the threat vectors $\boldsymbol{r}_{t/p}$ and $\boldsymbol{r}_{t/c}$ for the peripheral and central vision cameras, respectively. The intersection $\boldsymbol{r}_{t/g}$ of the threat in the global reference frame is then

$$\boldsymbol{r}_{t/g} = \boldsymbol{r}_{p/g} + \frac{\|\boldsymbol{r}_{t/c} \times \boldsymbol{r}_{p/c}\|}{\|\boldsymbol{r}_{t/c} \times \boldsymbol{r}_{t/p}\|}\boldsymbol{r}_{t/p} \tag{12}$$

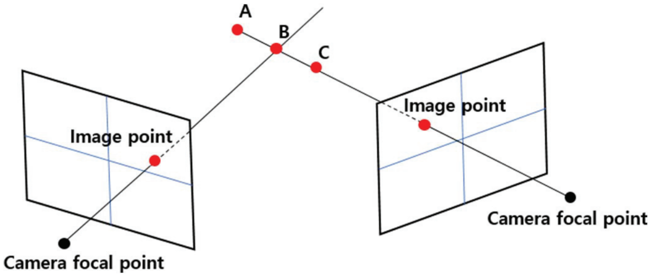


Fig. 11 Stereo vision.

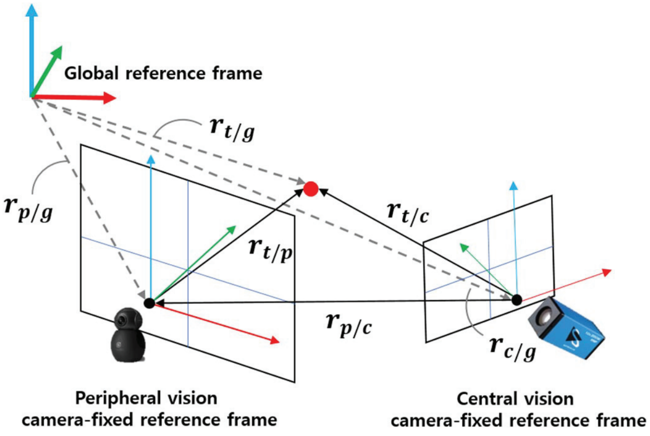

Global reference frame
$r_{t/g}$
$r_{p/g}$
$r_{t/c}$
$r_{t/p}$
$r_{p/c}$
$r_{c/g}$
Peripheral vision camera-fixed reference frame
Central vision camera-fixed reference frame

Fig. 12 Geometry of threat localization.

where $\boldsymbol{r}_{p/g}$ and $\boldsymbol{r}_{p/c}$ represent the optical center position of the peripheral vision camera with respect to the global frame and the central vision camera frame, respectively. [All vectors in Eq. (12) are assumed to be expressed in the global frame.]

If two 3D threat vectors intersect, the threat position can be easily estimated using Eq. (12). However, several sources of error ensure that two threat vectors will rarely intersect. Following is an incomplete list:

1) *Lens distortion:* The effect of light refraction through the camera lens can be partially corrected through camera calibration, but cannot be removed entirely.

2) *Camera pose error:* To compute the intersection point of threat vectors, the threat vectors should be expressed in a common frame (e.g., the global frame). The position and orientation of each camera frame in the global frame are obtained from an IMU and a GPS sensor attached to each camera, but these measurements are imperfect.

3) *Feature correspondence:* The point on the threat that determines the threat vector for one camera, a point that is determined using a feature detection algorithm (color detection, corner detector, etc.), may not correspond to the same detected feature point in the other camera image.

Approaches have been suggested to address the nonintersection of threat vectors [43–45]. The midpoint method computes the shortest line connecting two 3D vectors and takes the midpoint of the connecting line segment as an estimate for the intersection. This method is relatively easy to implement and fast to compute. However, if the error in the threat vectors is large, the localization error in the stereo ranging method is also large. Moreover, the midpoint method occasionally computes a (nonphysical) negative range [46]. An alternative, known as the optimal method, corrects the two 3D vectors based on the epipolar constraint that is satisfied when two 3D vectors are on a common plane (the epipolar plane). The two corrected threat vectors necessarily intersect. Even after the correction, there could be some reprojection error generated by the correction process. Mandun et al. [44] analyzed this reprojection error of the stereo vision system against the error from each camera. The reprojection error contributes to the threat localization error depending on the system parameters of the stereo vision system (baseline, target distance, etc.). Fooladgar et al. [43] discussed the uncertainty in different operating conditions. We consider this localization error in Secs. IX and X. For the implementation in this paper, we used the optimal method to correct the threat vectors because the optimal method gives lower localization error than the midpoint method.

VIII. System Architecture and Data Acquisition

The algorithms and hardware explained in the previous sections are finally integrated into a PCV system to generate a dataset. In this section, we describe the architecture of the PCV system and the data generation setup. The dataset is generated using ground-based and air-based PCV systems with various types of mock threats. The pros and cons of each type of the PCV system are described here. We also describe a track management strategy for managing multiple threats within the peripheral camera's FOV.

A. Data Acquisition

Figure 13 depicts the experimental setup for the dataset in which a mock threat aircraft streams its position to a ground station, providing ground truth to assess the localization strategy. For the PCV system, the GPS position of the system and the imagery from the two cameras are acquired and processed on an Nvidia TX2. The hardware and software setup is based on the ROS framework. After processing the data, the pointing cue is sent to the gimbal for the central vision camera. All data are stored in the *rosbag* file format.

Experiments were performed using two PCV configurations (ground-based and air-based) and various types of mock threats. In a given experiment, a threat maneuvers within the detectable range (100 m for the current system specifications) and the PCV system observes and records the data on the ground or while hovering in the air. Mock threats included a human, a quadcopter, a fixed-wing unmanned aerial vehicle

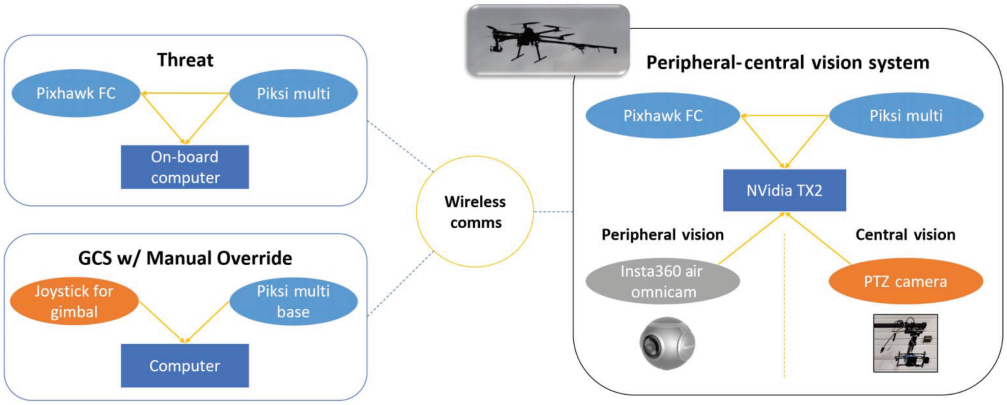


Fig. 13 Setup for generating image datasets.

Table 2 PCV system dataset

Host type	Host quantity	Threat type	Time duration, min
Ground-based	One	Human	20
		Multirotor UAV	4
Air-based	One	Multirotor UAV	20
		Fixed-wing UAV	19
Air-based	Two	Multirotor UAV	30

(UAV), and a manned aircraft. Table 2 shows the details of the dataset that was gathered.

Advantages of the ground-based PCV system include unlimited power, low position error for the camera system, and clear, steady imagery. For these and other reasons, the ground-based PCV system is easier to use and its data are easier to process, because there is no need to stabilize the imagery. It is not surprising that more than 70% of commercial counter-UAS systems are ground-based [47].

The primary disadvantage of a ground-based PCV system is its inability to vary the camera perspective. For some threats in the counter-UAS application, the ability to induce particular relative motions by maneuvering the host could aid detectability and localization accuracy. Moreover, a threat detection and tracking system developed for use on a small unmanned aircraft can serve the dual purpose of ABDAA. One of the virtues of the PCV system is the low SWaP-C that enables it to be easily integrated into a small UAS. The performance of the air-based system, on the other hand, is affected by GPS accuracy, image quality from the moving platform, and battery life.

One issue for both single PCV system configurations that were tested is the short baseline (2 m) between the central vision camera and the peripheral vision camera. Because the PCV system estimates the threat position based on triangulation, as shown in Sec. VII, a shorter baseline can increase the error in the threat vector error; this issue is discussed further in Sec. IX. To explore the opportunities afforded by larger baselines, we also constructed and used a second PCV system in parallel.

For the dataset collected using the ground-based system, two types of threat were used: a human and a quadcopter. For the air-based system, two types of threat were used: a quadcopter and a fixed-wing UAV. Although the central vision camera is physically stabilized by the gimbal, the peripheral vision camera was affected by vibration when obtaining data from the air-based system. The peripheral imagery was software stabilized as described in Sec. IV.A. The dataset includes imagery of the threat from two cameras, the host's GPS location, the threat's GPS location, and the gimbal orientation for the central vision camera. The dataset is used to test the detection algorithms and the localization algorithm. Figure 14 shows example imagery from the dataset.

B. Track Management

To this point, we have focused on the case of a single PCV system and a single threat. To expand its utility, the system should allow for multiple threats. Moreover, the system's capability can be improved by allowing for multiple sensing systems, such as additional ground- and air-based sensing systems [48,49]. Here, a multi-object tracking (MOT) solution to process measurements obtained from multiple cameras/sensors is developed to facilitate continuous threat monitoring even with patchy detection performance. The MOT uses established tracking algorithms, but the modular architecture allows one to replace component filtering algorithms, such as the extended Kalman filter that is currently used for state estimation of each track, with alternative algorithms as desired for development purposes. Development in ROS also supports modularity, allowing an arbitrary number of sensors, moving or stationary, with automatic handling of all coordinate frame conversions performed by the transform ROS package [50]. Sensor fusion between multiple sensors is accomplished using synchronized measurements in the update step. The tracker layout is shown in Fig. 15.

The detection preprocessor consolidates incoming measurements/detections from all sensors and transforms them from the sensor frame to the global frame. In the data association step, valid detections from the data preprocessor are first compared with the current track table. A Mahalanobis distance metric is used to associate incoming measurements with any existing tracks. Any unassociated measurement is initialized as a new track, though at least one additional detection is required before publishing the track, to reduce false tracks.

The Bayesian tracker, an extended Kalman filter (EKF) [51] based estimator, takes in the associated measurements and performs

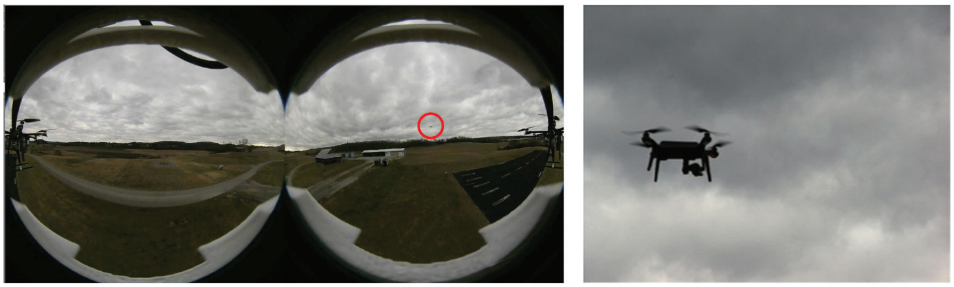


Fig. 14 Example dataset images: synchronous peripheral (left) and central (right) vision images.

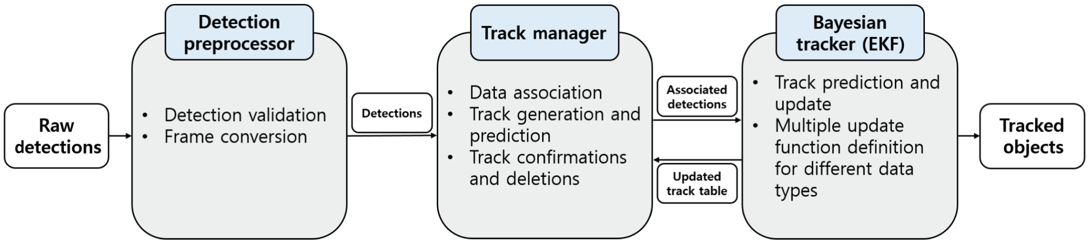


Fig. 15 Multi-object tracker layout.

the "update" step of the predict–update cycle. If a particular track does not have any associations, its next state is predicted using a constant acceleration motion model. In the case that multiple measurements are associated with a given track, all measurements are used to better estimate the state. The track manager then cycles through all existing tracks, pruning those whose covariance has grown larger than a user-defined maximum, and then forwards the remaining tracks to the PCV system scheduler that cues sensors, such as a central vision camera.

The design and performance of the data association and state estimation algorithms mentioned above are described in textbooks on multi-object tracking [52,53].

IX. System Analysis

The localization error from the heterogeneous stereo vision algorithm shown in Sec. VII is affected by several system parameters. In this section, we further analyze parameter effects on localization error using Monte Carlo simulation results.

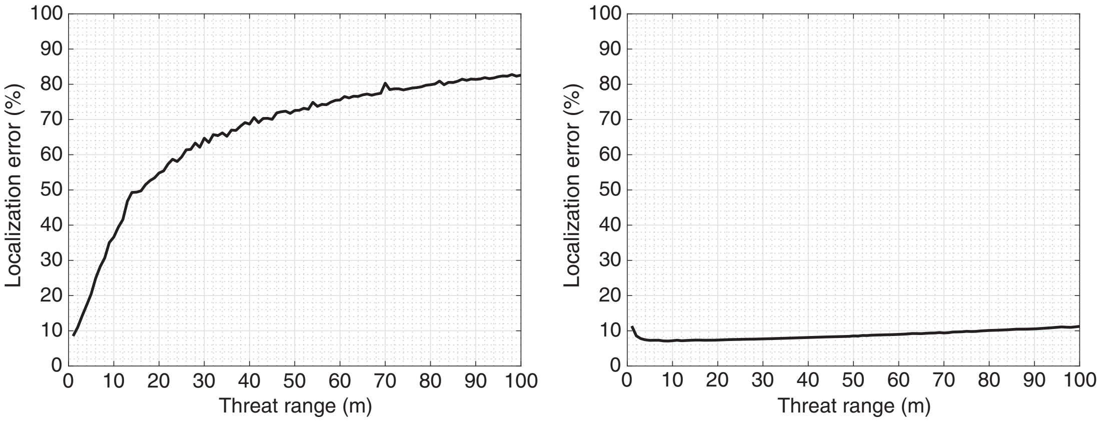
A. Localization Error Analysis

Numerous issues contribute to threat localization error, including feature extraction error, camera calibration, lens distortion, camera resolution, inertial sensor accuracy, and gimbal sensor accuracy as well as the length of the baseline between the two cameras. The baseline of the single PCV system is 2 m, and the minimum pixel coverage needed to detect a threat using the optical flow algorithm with peripheral vision imagery is about 25 px (5 px × 5 px). The minimum pixel coverage for detection using YOLOv3 with central vision imagery is about 900 px (30 px × 30 px). Even after undistorting peripheral camera imagery, the low relative pixel density near the edges of the image can prevent the detection of threats in these regions. The problem is compounded because the undistortion function automatically interpolates pixel values, backfilling gaps in regions with sparse pixel coverage. All these sources of error aggregate, resulting in an erroneous threat vector from which we may assess the localization error.

To investigate measurement uncertainty, Monte Carlo simulations were implemented using the system parameters shown in Table 1. Zero-mean Gaussian noise is superimposed on the threat vectors in the horizontal and vertical directions of the camera-fixed reference frame, corresponding to a 30 px standard deviation for the peripheral vision image and a 3 px standard deviation for the central vision image. This synthetic error was tuned empirically to closely match that of the actual hardware. (We neglect error associated with feature extraction.) Localization error estimates with the random Gaussian noise were obtained from 10,000 samples and averaged. The trends observed in these 10,000 samples were evident in the first 1000 samples, suggesting that 10,000 samples are sufficient. Multiple of these Monte Carlo simulations were conducted by varying the threat range. Figure 16 shows the localization error using a 2 m baseline and a 100 m baseline, as obtained from Monte Carlo simulations. The localization error increases with distance to the threat, similar to the experimental results in Fig. 17. As shown in Figs. 17 and 16, the localization error can be reduced by increasing the baseline of the PCV system. In the following section, a system performance analysis is conducted to see the relationship between system parameters, such as camera position and resolution, and localization accuracy.

B. System Performance Analysis

Although one may assess the overall performance of a multi-component, multi-algorithm system for detection, tracking, and classification, the meaning of such an assessment is difficult to interpret given the number of parameters involved in configuring such a system. It is generally easier to assess the performance of individual components and to compare each with comparable existing approaches based on the particular role they address. That said, some components are easier to assess than others. As described in Sec. VI.B, for example, classification performance is difficult to measure without a sufficiently large, diverse dataset. Performance of the off-the-shelf tracking solution used here can be readily measured, but the issue has been explored elsewhere. In any case, tracking performance is closely tied to detection and localization quality. Localization error is affected by a number of system parameters, including the camera baseline and resolution. The quality of the peripheral vision camera, in particular, dominates the performance of the PCV system described here. To compare the PCV system performance using different system parameters, the localization error is computed from Monte Carlo simulations using various values for camera location and peripheral vision camera resolution.



a) Localization error using 2 m baseline

b) Localization error using 100 m baseline

Fig. 16 Localization error versus threat range from Monte Carlo simulation.

1. *Camera Location*

A formula is given in [54] relating triangulation-based localization error and sensor error:

$$\delta_r = \frac{\|\boldsymbol{r}_{t/p}\|\|\boldsymbol{r}_{t/c}\|}{\sin\theta}\delta_s \tag{13}$$

where δ_r is the magnitude of the localization error, δ_s is the magnitude of threat vector error, and θ is the intersection angle between two threat vectors:

$$\theta = \cos^{-1}\left(\frac{\boldsymbol{r}_{t/p}\cdot\boldsymbol{r}_{t/c}}{\|\boldsymbol{r}_{t/p}\|\|\boldsymbol{r}_{t/c}\|}\right)$$

Equation (13) implies that a shorter range to the threat and a nearly orthogonal viewing angle ensure the smallest localization error. Figure 18a shows the localization error obtained from the Monte Carlo simulations in which the threat range is fixed at 50 m but the intersection angle θ is varied. As shown in the plots, localization error decreases with larger intersection angles, which helps to explain why the longer baseline improves localization accuracy in Figs. 17 and 16. Figure 18b shows the localization error with a fixed intersection angle $\theta = 90°$ and varying range to the threat. The localization error increases linearly with increasing range, at about 8 cm per meter. Analysis indicates that both threat range and intersection angle θ are important determinants of localization accuracy, but θ appears to play the more important role. Accuracy degrades quickly for threat vector intersection angles less than about 30°.

2. *Peripheral Vision Camera Resolution*

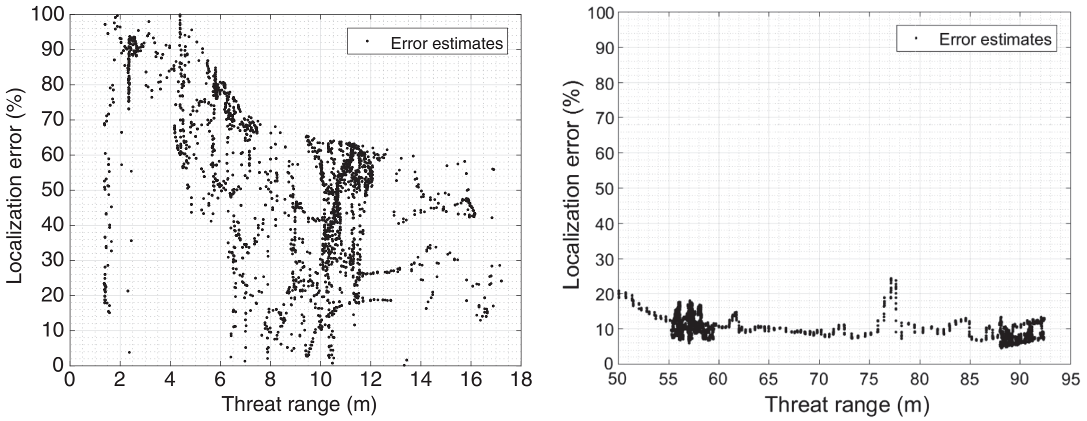
The peripheral vision camera has a lower resolution than the central vision camera because of the wide FOV (see Fig. 14). The PCV system localization performance is thus more affected by the peripheral vision camera resolution than that of the central vision camera. Recalling that localization from a single image is impossible, because of the range ambiguity, it is intuitive that system performance would be limited by the lower resolution camera.

We note that the absolute localization error indicated in these performance analysis results is partly a consequence of system architecture choices unrelated to optical performance, such as the need for a low SWaP-C system that is compatible with the ROS computing framework. The emphasis here is on the relative effects of various system parameters, rather than on absolute performance, which could easily be improved by using higher quality optics.

The maximum detectable range of the current system for a threat that is 1 m wide is 100 m. Figures 19a and 19b illustrate the maximum detectable range and range error, respectively, along with the peripheral vision camera resolution. As shown in Fig. 19a, the maximum detectable range increases almost linearly with resolution, which means that a higher resolution peripheral vision camera enables detection of more distant threats. Each plot in Fig. 19b indicates the localization error for several example threat ranges. As shown in the figure, the localization error decreases with increasing camera resolution, but a “knee” is observed, which indicates a diminishing return beyond roughly 5 MP. Considering that the resolution of the current peripheral vision camera is 2.26 MP, increasing the peripheral vision camera’s resolution to 5 MP would be a reasonable next step for improving system performance.

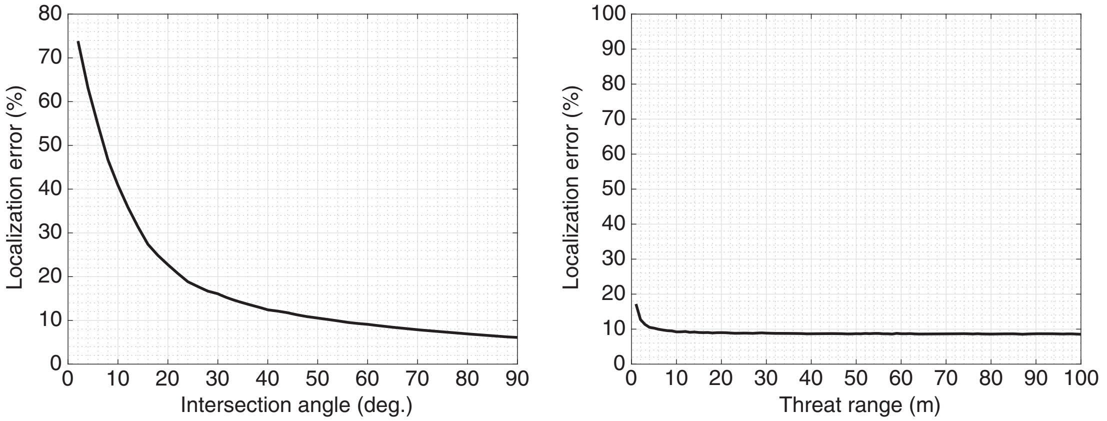
X. Threat Localization Results

The heterogeneous stereo vision algorithm is implemented in ROS-based software by replaying the rosbag dataset mentioned in Sec. VIII. In this section, the threat localization error using the actual dataset is estimated to see if the actual estimation error coincides with



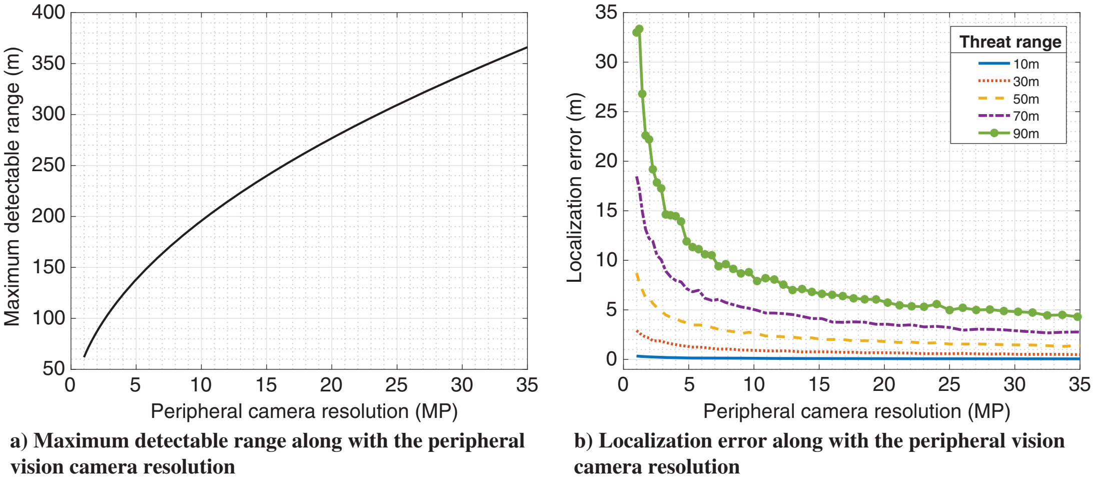
a) Localization error using single PCV system b) Localization error using two PCV systems

Fig. 17 Threat localization error of the dataset.



a) Localization error with constant threat range (50 m) b) Localization error with constant intersection angle (90°)

Fig. 18 Localization error using Monte Carlo simulation.



a) Maximum detectable range along with the peripheral vision camera resolution

b) Localization error along with the peripheral vision camera resolution

Fig. 19 The effect of the peripheral vision camera resolution.

the analysis in Sec. IX. Figure 20 shows a screenshot of the output from the ROS-based threat localization software. Once the threat is detected in both camera images, threat vectors from the two cameras are generated (the rays originating from the two camera frames in Fig. 20), and the intersection point of two threat vectors is computed to estimate the threat's position. The threat in the dataset has its own GPS, for ground truth, and this independent position measurement is compared with the estimated position to check the localization performance.

Figure 17a depicts the localization error (an absolute distance error between the ground truth threat position and the estimated threat position) versus the range of the threat using a single PCV system. As shown in the figure, the localization error and the error variation is quite large. One major source of error in the threat vectors is the feature extraction error. The feature extraction algorithms applied to both camera images are not guaranteed to detect the threat. If one of the cameras detects the threat incorrectly, the localization error is large, as reflected in Fig. 17a. The second source of error is the difference in camera resolutions. The peripheral vision camera and the central vision camera have different camera models: a pinhole camera model and a fisheye camera model, respectively. This difference generates more error than would arise using identical cameras. This localization error is exacerbated by a short baseline relative to threat range, as mentioned in Sec. IX. Shah and Aggarwal [55] estimated range error using stereo fisheye lenses for nearby objects (5–8 m distant) and found that the error was 8–10%. Lipnickas and Knyš [56] used a perspective stereo vision camera system for objects at 1–4 m distant and found that the error was 1–4%. Nedevschi et al. [57,58] used a perspective stereo camera system for more distant objects (10–95 m) and found that the error was 1–2%. These results are much better than the results of Fig. 17a because the researchers used identical cameras in their stereo vision systems and the cameras were placed in stable locations (e.g., on the ground or a ground vehicle), not in the air. We note that the large error and error variation shown in Fig. 16a decrease with the longer baseline. Figure 17b shows the localization error using two PCV systems, where the camera baseline is 90 m. In this figure, we still see some error variation due to feature extraction error, but because of the greater range to the threat, the localization error is reduced to around 10%, much less than for the single PCV system. Also, the localization error does not increase as rapidly with increasing range compared with the single PCV system. The results in Fig. 17 closely coincide with those in Fig. 16 in Sec. IX, affirming the well-known fact that a longer baseline improves triangulation accuracy. Ongoing work involves the coordinated use of multiple ground- and air-based PCV systems for counter-UAS and cooperative ABDAA applications.

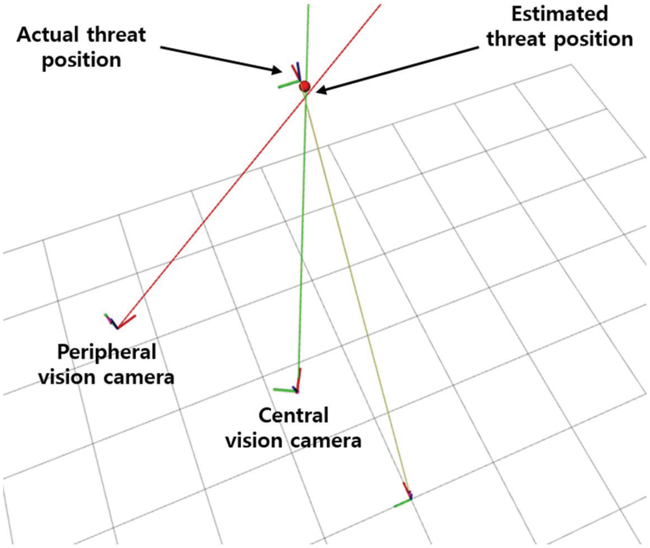


Fig. 20 Threat localization.

XI. Conclusions

This paper describes a PCV system to detect, localize, and classify threats. The peripheral vision camera initially detects the threat using optical flow, after image undistortion and stabilization. The threat bearing in the peripheral vision camera-fixed reference frame is then transformed into the central vision camera-fixed reference frame with an optimized assumed range. The central vision camera is then cued to slew toward the threat. Once the threat is acquired by the second camera, the range is estimated using a heterogeneous stereo vision algorithm and the assumed range is replaced with this estimate. The refined range estimate is then used to compute the optimal zoom value. The more detailed image of the threat available from the central vision camera allows one to classify the threat using a DNN and also to estimate the pose of the threat.

To assess the threat localization performance, an experimental dataset was generated using a variety of mock threats. Results show that the localization accuracy is quite limited using the current low-cost cameras in the given configuration. Analysis of localization error for the experimental dataset obtained using a single PCV system revealed a large localization error with large variability. The large error variation is due to error in the threat vectors, for which the major contributor is feature extraction error. It was also observed, however, that localization error decreases substantially with a longer baseline as obtained in experiments using two PCV systems. Monte Carlo simulations allowed further investigation of the effect of system parameters on the localization error. The results indicate that for threat vector intersection angles smaller than about 30°, localization error increases rapidly. The short baseline configuration of a single PCV system places a fundamental limit on stereo ranging accuracy, but multiple

PCV systems operating in concert can provide much more accurate range estimates. This accuracy is also influenced, however, by camera quality and performance of the feature extraction algorithm that helps to define the threat vector. Ongoing efforts are aimed at developing and testing algorithms for the coordinated use of multiple PCV systems to improve threat detection and localization performance.

Acknowledgment

The authors gratefully acknowledge the support of the Center for Unmanned Aircraft Systems, a National Science Foundation (NSF) Industry/University Cooperative Research Center (I/UCRC) under NSF Grant No. CNS-1650465.

References

[1] Kang, C., Davis, J., Woolsey, C. A., and Choi, S., "Sense and Avoid Based on Visual Pose Estimation for Small UAS," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, New York, Sept. 2017, pp. 3473–3478. https://doi.org/10.1109/IROS.2017.8206188

[2] Kang, C., Chaudhry, H., Woolsey, C. A., and Kochersberger, K. B., "Development of a Peripheral-Central Vision System for Small UAS Tracking," *AIAA SciTech*, AIAA Paper 2019-2074, Jan. 2019. https://doi.org/10.2514/6.2019-2074

[3] Drulea, M., Szakats, I., Vatavu, A., and Nedevschi, S., "Omnidirectional Stereo Vision Using Fisheye Lenses," *2014 IEEE International Conference Intelligent Computer Communication and Processing (ICCP)*, IEEE, New York, Sept. 2014, pp. 251–258. https://doi.org/10.1109/ICCP.2014.6937005

[4] Kita, N., and Kita, Y., "Reference Plane Based Fisheye Stereo Epipolar Rectification," *12th International Conference on Computer Vision Theory and Applications (VISAPP)*, Feb.–March 2017, pp. 308–320. https://doi.org/10.5220/0006261003080320

[5] Chen, C. H., Yao, Y., Page, D., Abidi, B., Koschan, A., and Abidi, M., "Heterogeneous Fusion of Omnidirectional and PTZ Cameras for Multiple Object Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 8, 2008, pp. 1052–1063. https://doi.org/10.1109/TCSVT.2008.928223

[6] Yu, M. S., Wu, H., and Lin, H. Y., "A Visual Surveillance System for Mobile Robot Using Omnidirectional and PTZ Cameras," *SICE Annual Conference 2010*, IEEE, New York, Aug. 2010, pp. 37–42.

[7] Cui, Y., Samarasckera, S., Huang, Q., and Greiffenhagen, M., "Indoor Monitoring via the Collaboration Between a Peripheral Sensor and a Foveal Sensor," *1998 IEEE Workshop on Visual Surveillance*, IEEE, New York, Jan. 1998, pp. 2–9. https://doi.org/10.1109/WVS.1998.646014

[8] Scotti, G., Marcenaro, L., Coelho, C., Selvaggi, F., and Regazzoni, C. S., "Dual Camera Intelligent Sensor for High Definition 360 Degrees Surveillance," *IEE Proceedings-Vision, Image and Signal Processing*, Vol. 152, No. 2, 2005, pp. 250–257. https://doi.org/10.1049/ip-vis:20041302

[9] Iraqui, A., Dupuis, Y., Boutteau, R., Ertaud, J. Y., and Savatier, X., "Fusion of Omnidirectional and PTZ Cameras for Face Detection and Tracking," *2010 International Conference on Emerging Security Technologies*, IEEE, New York, Sept. 2010, pp. 18–23. https://doi.org/10.1109/EST.2010.16

[10] Fahn, C. S., and Lo, C. S., "A High-Definition Human Face Tracking System Using the Fusion of Omni-Directional and PTZ Cameras Mounted on a Mobile Robot," *2010 5th IEEE Conference on Industrial Electronics and Applications*, IEEE, New York, June 2010, pp. 6–11. https://doi.org/10.1109/ICIEA.2010.5514985

[11] Baris, I., and Bastanlar, Y., "Classification and Tracking of Traffic Scene Objects with Hybrid Camera Systems," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, New York, Oct. 2017, pp. 1–6. https://doi.org/10.1109/ITSC.2017.8317588

[12] Muñoz-Salinas, R., Medina-Carnicer, R., Madrid-Cuevas, F. J., and Carmona-Poyato, A., "Particle Filtering with Multiple and Heterogeneous Cameras," *Pattern Recognition*, Vol. 43, No. 7, 2010, pp. 2390–2405. https://doi.org/10.1016/j.patcog.2010.01.015

[13] Eynard, D., Vasseur, P., Demonceaux, C., and Frémont, V., "UAV Altitude Estimation by Mixed Stereoscopic Vision," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, New York, Oct. 2010, pp. 646–651. https://doi.org/10.1109/IROS.2010.5652254

[14] Eynard, D., Demonceaux, C., Vasseur, P., and Fremont, V., "UAV Motion Estimation Using Hybrid Stereoscopic Vision," *2011 IAPR Conference on Machine Vision Applications (MVA)*, Springer, Berlin, June 2011, pp. 340–343.

[15] Eynard, D., Vasseur, P., Demonceaux, C., and Frémont, V., "Real Time UAV Altitude, Attitude and Motion Estimation from Hybrid Stereovision," *Autonomous Robots*, Vol. 33, Nos. 1–2, 2012, pp. 157–172. https://doi.org/10.1007/s10514-012-9285-0

[16] Lee, J. J., and Kim, G., "Robust Estimation of Camera Homography Using Fuzzy RANSAC," *International Conference on Computational Science and Its Applications*, Springer, Berlin, Aug. 2007, pp. 992–1002. https://doi.org/10.1007/978-3-540-74472-6_81

[17] Monnin, D., Bieber, E., Schmitt, G., and Schneider, A., "An Effective Rigidity Constraint for Improving RANSAC in Homography Estimation," *International Conference on Advanced Concepts for Intelligent Vision Systems*, Springer, Berlin, Dec. 2010, pp. 203–214. https://doi.org/10.1007/978-3-642-17691-3_19

[18] Lusk, P. C., and Beard, R. W., "Visual Multiple Target Tracking from a Descending Aerial Platform," *2018 Annual American Control Conference (ACC)*, IEEE, New York, June 2018, pp. 5088–5093. https://doi.org/10.23919/ACC.2018.8431915

[19] Kang, C., and Woolsey, C. A., "Scheduled Imaging of Multiple Threat Aircraft Using Modified Traveling Salesman Problem" (to be published).

[20] Miranda, S., Baker, C., Woodbridge, K., and Griffiths, H., "Knowledge-Based Resource Management for Multifunction Radar," *IEEE Signal Processing Magazine*, Vol. 23, No. 1, 2006, pp. 66–76. https://doi.org/10.1109/MSP.2006.1593338

[21] Miranda, S. L., Baker, C. J., Woodbridge, K., and Griffiths, H. D., "Simulation Methods for Prioritizing Tasks and Sectors of Surveillance in Phased Array Radar," *International Journal of Simulation*, Vol. 5, Nos. 1–2, 2004, pp. 18–25.

[22] Miranda, S. L. C., Baker, C. J., Woodbridge, K., and Griffiths, H., "Fuzzy Logic Approach for Prioritisation of Radar Tasks and Sectors of Surveillance in Multifunction Radar," *IET Radar, Sonar and Navigation*, Vol. 1, No. 2, 2007, pp. 131–141. https://doi.org/10.1049/iet-rsn:20050106

[23] Scaramuzza, D., Martinelli, A., and Siegwart, R., "A Toolbox for Easily Calibrating Omnidirectional Cameras," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, New York, Oct. 2006, pp. 5695–5701. https://doi.org/10.1109/IROS.2006.282372

[24] Horn, B. K., and Schunck, B. G., "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, Nos. 1–3, 1981, pp. 185–203. https://doi.org/10.1016/0004-3702(81)90024-2

[25] Lucas, B. D., and Kanade, T., "An Iterative Image Registration Technique with an Application to Stereo Vision," *7th International Joint Conference on Artificial Intelligence (IJCAI)*, Morgan Kaufmann Publ., Burlington, MA, Aug. 1981, pp. 674–679.

[26] Harris, C., and Stephens, M., "A Combined Corner and Edge Detector," *Alvey Vision Conference*, Vol. 15, British Machine Vision Assoc., Durham, U.K., 1988, pp. 10–5244.

[27] Fischler, M. A., and Bolles, R. C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395. https://doi.org/10.1145/358669.358692

[28] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME—Journal of Basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45. https://doi.org/10.1115/1.3662552

[29] Lowe, D. G., "Local Feature View Clustering for 3-D Object Recognition," *Proceedings of 2001 IEEE Conference on Computer Vision Pattern Recognition, CVPR 2001*, Vol. 1, IEEE, New York, 2001, p. 682. https://doi.org/10.1109/CVPR.2001.990541

[30] Bay, H., Tuytelaars, T., and Van Gool, L., "Surf: Speeded Up Robust Features," *European Conference on Computer Vision*, Springer, Berlin, May 2006, pp. 404–417. https://doi.org/10.1007/11744023_32

[31] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: An Efficient Alternative to SIFT or SURF," *International Conference on Computer Vision*, IEEE, New York, Nov. 2011, pp. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

[32] Henriques, J. F., Caseiro, R., Martins, P., and Batista, J., "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 3, 2014, pp. 583–596. https://doi.org/10.1109/TPAMI.2014.2345390

[33] Babenko, B., Yang, M. H., and Belongie, S., “Robust Object Tracking with Online Multiple Instance Learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 8, 2010, pp. 1619–1632.
https://doi.org/10.1109/TPAMI.2010.226

[34] Lin, J., Ji, X., Xu, W., and Dai, Q., “Absolute Depth Estimation from a Single Defocused Image,” *IEEE Transactions on Image Processing*, Vol. 22, No. 11, 2013, pp. 4545–4550.
https://doi.org/10.1109/TIP.2013.2274389

[35] Rajagopalan, A. N., and Chaudhuri, S., “An MRF Model-Based Approach to Simultaneous Recovery of Depth and Restoration from Defocused Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 7, 1999, pp. 577–589.
https://doi.org/10.1109/34.777369

[36] Hiura, S., and Matsuyama, T., “Depth Measurement by the Multi-Focus Camera,” *1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, New York, June 1998, pp. 953–959.
https://doi.org/10.1109/CVPR.1998.698719

[37] “Visual Aircraft Recognition TC 3-01.80,” Department of the Army, Army Publishing Directorate TR TC 3-01.80, May 2017, https://armypubs.army.mil.

[38] Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., and Birchfield, S., “Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, New York, June 2018, pp. 969–977.
https://doi.org/10.1109/CVPRW.2018.00143

[39] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv preprints, 2017.

[40] Redmon, J., and Farhadi, A., “YOLO9000: Better, Faster, Stronger,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, New York, 2017, pp. 7263–7271.
https://doi.org/10.1109/cvpr.2017.690

[41] Redmon, J., and Farhadi, A., “YOLOv3: An Incremental Improvement,” arXiv preprint, 2018.

[42] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., “Microsoft Coco: Common Objects in Context,” *European Conference on Computer Vision (ECCV)*, Springer, Berlin, Sept. 2014, pp. 740–755.
https://doi.org/10.1007/978-3-319-10602-1_48

[43] Fooladgar, F., Samavi, S., Soroushmehr, S. M. R., and Shirani, S., “Geometrical Analysis of Localization Error in Stereo Vision Systems,” *IEEE Sensors Journal*, Vol. 13, No. 11, 2013, pp. 4236–4246.
https://doi.org/10.1109/JSEN.2013.2264480

[44] Mandun, Z., Lichao, Q., Guodong, C., and Ming, Y., “A Triangulation Method in 3D Reconstruction from Image Sequences,” *2009 Second International Conference on Intelligent Networks and Intelligent Systems*, IEEE, New York, Nov. 2009, pp. 306–308.
https://doi.org/10.1109/ICINIS.2009.84

[45] Kanatani, K., Sugaya, Y., and Niitsuma, H., “Triangulation from Two Views Revisited: Hartley-Sturm vs. Optimal Correction,” *19th British Machine Vision Conference (BMVC 2008)*, British Machine Vision Assoc., Durham, U.K., Sept. 2008, pp. 173–182.
https://doi.org/10.5244/c.22.18

[46] Lee, S. H., and Civera, J., “Triangulation: Why Optimize?” *30th British Machine Vision Conference 2019, (BMVC) 2019*, British Machine Vision Assoc., Durham, U.K., Sept. 2019, p. 162.

[47] Michel, A. H., *Counter-Drone Systems*, Center for the Study of the Drone at Bard College, Annandale-on-Hudson, NY, 2019, https://dronecenter.bard.edu/files/2018/02/CSD-Counter-Drone-Systems-Report.pdf.

[48] Sahawneh, L. R., Wikle, J. K., Kaleo Roberts, A., Spencer, J. C., McLain, T. W., Warnick, K. F., and Beard, R. W., “Ground-Based Sense-and-Avoid System for Small Unmanned Aircraft,” *Journal of Aerospace Information Systems*, Vol. 15, No. 8, 2018, pp. 501–517.
https://doi.org/10.2514/1.I010627

[49] Tolman, S., and Beard, R. W., “Counter UAS Using a Formation Controlled Dragnet,” *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, New York, June 2017, pp. 1665–1672.
https://doi.org/10.1109/ICUAS.2017.7991391

[50] “tf2—ROS Wiki,” 2021, http://wiki.ros.org/tf2.

[51] Smith, G. L., Schmidt, S. F., and McGee, L. A., *Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle*, NASA, Washington, D.C., 1962, pp. 5–8.

[52] Challa, S., Morelande, M. R., Mušicki, D., and Evans, R. J., *Fundamentals of Object Tracking*, Cambridge Univ. Press, Cambridge, England, U.K., 2011, Chaps. 2, 4, 5.
https://doi.org/10.1017/CBO9780511975837

[53] Bar-Shalom, Y., and Li, X. R., *Estimation with Applications to Tracking and Navigation*, Wiley, Hoboken, NJ, 2001, Chap. 10.

[54] Kelly, A., “Precision Dilution in Triangulation Based Mobile Robot Position Estimation,” *Intelligent Autonomous Systems*, Vol. 8, June 2003, pp. 1046–1053.

[55] Shah, S., and Aggarwal, J. K., “Depth Estimation Using Stereo Fish-Eye Lenses,” *1st International Conference on Image Processing*, IEEE, New York, Nov. 1994, pp. 740–744.
https://doi.org/10.1109/ICIP.1994.413669

[56] Lipnickas, A., and Knyš, A., “A Stereovision System for 3-D Perception,” *Elektronika ir Elektrotechnika*, Vol. 91, No. 3, 2009, pp. 99–102.

[57] Nedevschi, S., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C., Schmidt, R., and Graf, T., “High Accuracy Stereo Vision System for Far Distance Obstacle Detection,” *IEEE Intelligent Vehicles Symposium, 2004*, IEEE, New York, June 2004, pp. 292–297.
https://doi.org/10.1109/IVS.2004.1336397

[58] Nedevschi, S., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C., Schmidt, R., and Graf, T., “Stereovision Approach for Obstacle Detection on Non-Planar Roads,” *1st International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, IEEE, New York, Aug. 2004, pp. 11–18.
https://doi.org/10.5220/0001139300110018

M. J. Kochenderfer
Associate Editor