JOURNAL OF AEROSPACE INFORMATION SYSTEMS

Technical Notes

Scheduled Imaging of Multiple Threat Aircraft Using a Modified Traveling Salesman Problem

Changkoo Kang* and Craig A. Woolsey†
Virginia Tech, Blacksburg, Virginia 24061

https://doi.org/10.2514/1.I010871

I. Introduction

IN DECEMBER 2018, a small unmanned aerial system (UAS) was observed at Gatwick airport in the United Kingdom, and this caused a major disruption at the airport. After this "drone chaos," Gatwick airport spent 6 million dollars to install a counter-UAS (C-UAS) system [1]. With the rapid proliferation of small UAS, the risk of accidents caused by small UAS is also growing [2]. Given this concern, research on C-UAS technology has immediate relevance. These systems use a variety of sensors such as electro-optical/infrared (EO/IR) cameras, RADAR, and acoustic sensors, and the data from these sensors are fused to detect and track a given threat [3]. According to [1], there are 323 commercial UAS detection systems in the world, and half of them use combined sensor systems.

Omnidirectional–directional combined sensor systems such as radar–camera systems [4], dual-camera systems [5–10], and lidar–camera systems [11,12] are common types of combined sensor systems. In our previous work [13], an air-based peripheral–central vision (PCV) system to detect and characterize airborne threats was developed. The PCV system is a heterogeneous stereo vision system comprising an omnidirectional peripheral vision camera and a directional central vision camera capable of pan-tilt-zoom (PTZ) operation. The peripheral vision camera provides continuous visual coverage of the environment for threat detection, although with relatively low and nonuniform resolution. The central vision camera complements the peripheral vision camera by providing a high-resolution image when cued to observe a threat. An extended version of this concept is discussed in [14], where the authors develop a decentralized multitarget state estimation scheme to allocate a number of remote central vision sensors. In the system described in [15], the pair of cameras affords the opportunity to use stereo vision for estimating the 3D position of the threat while the high-resolution image of the central vision camera allows monocular vision-based estimation of the threat aircraft attitude. Thus, one may obtain both the position and the attitude of the threat aircraft using the PCV system. However, in [13], only a single threat has been assumed, and this assumption may not be reasonable for an urban environment or more crowded areas in which multiple objects may exist in the air.

Here, we consider an environment with multiple small UAS threats. The omnidirectional sensor is able to observe all these threats at once using its large field-of-view (FOV). The directional sensor then slews to observe each threat to obtain detailed data (e.g., position, velocity, aircraft classification). A threat that is rapidly approaching may need to be observed as soon as possible. However, all of these threats keep moving and changing their state. To keep the threat data updated, the directional sensor needs to keep maneuvering to observe each threat according to a particular "threat schedule" determined by a risk analysis based on the state of threats such as the approach speed and distance to the threat.

Threat scheduling algorithms are commonly used with phased array radars. The phased array radar is capable of tracking multiple threats at the same time, but the operating cost is high because of power requirements. Threat scheduling algorithms have therefore been studied for efficient radar resource management, where the sensor focuses on those threats that pose the greatest risk. Miranda et al. defined detailed criteria for prioritizing threats using fuzzy logic [16–18], an approach that was also used by Ding and Moo [19]. Other studies have proposed optimal threat scheduling algorithms for multiple PTZ cameras to minimize the uncertainty of threat data using information theory [20–23]. Other closely related research efforts have focused on fusing data from multiple sensors, both active and passive, to schedule and track multiple threats [10,24,25]. The Drone Net program [10,24] is especially relevant for combining all-sky and perspective camera imagery, among other ground-based sensors. Khosla et al. [4] proposed a novel threat scheduling algorithm for a radar-cued camera system. They prioritized threats based on time elapsed from the initial detection, FOV of the camera, and the slew angle required for the PTZ camera to image a given threat.

The threat scheduling algorithms mentioned above prioritize the risk posed by each threat over other concerns. These scheduling algorithms are especially appropriate for ground-based sensing systems, which are less limited by size, weight, power, and cost (SWaP-C) and which generally have a long sensing range. Scheduling algorithms for these systems do not need to consider the time taken for a secondary directional sensor to move toward threats (e.g., panning/tilting time for a PTZ camera), because the threats can be detected from a long stand-off distance. (For example, the sensing range of the ground-based radar considered in [26,27] is more than 3 km.) On the other hand, if the sensing range of the omnidirectional sensor is short (e.g., the sensing range of the PCV system for 1-m-size threat is 100 m [13]), the time that it takes for the directional sensor to maneuver and image a particular threat is crucial because one of the threats may harm the host system even before it is observed by directional sensor.

Time-efficient observation involves minimizing the observation time, that is, the total time required to observe all of the given threats using a directional sensor. Nevas and Proença [28] and Del Bimbo and Pernici [29] proposed PTZ camera scheduling algorithms that use Markov random fields (MRFs) and the traveling salesman problem (TSP) to minimize time to observe all targets. These two algorithms minimize the time to observe given threats; however, the risk posed by the threats is not considered for scheduling. For threat scheduling, both the slewing time required by the directional sensor and the risk posed by a given threat should be considered in determining the imaging order. If a threat that is near the current FOV of the directional sensor poses a relatively small risk, for example, the imaging and characterization of this threat might be postponed while the sensor images a higher priority threat. If it is possible to image this lower risk threat without significantly delaying acquisition of the higher risk threat, however, then doing so would be more time-efficient. A suitably designed scheduling algorithm should enable the system to observe all of the threats faster and/or more often.

We present a modified TSP formulation to schedule the imaging of multiple threats using a directional, narrow FOV imaging sensor

---

Received 10 June 2020; revision received 16 December 2020; accepted for publication 9 April 2021; published online 3 June 2021.

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

†Professor, Department of Aerospace and Ocean Engineering. Associate Fellow AIAA.

together with a fixed, wide FOV sensor. The target application is a passive, airborne sensing system to detect and track small UAS. The scheduling method weighs the risk posed by each threat against the time required to image the threat according to a criterion determined by the user. The performance of the proposed algorithm is studied using simulations where the relative priority between risk and timeliness is varied from one extreme to the other. The paper is organized as follows. Section II reviews the motivating application: an airborne, heterogeneous stereo vision system that can be used to detect and track small UAS. Section III introduces a method for assigning a risk value to each detected threat. Section IV presents the modified TSP whose solution is an "optimal tour" by which the directional sensor may service every threat. The complete threat scheduling algorithm design is described in Sec. V. Section VI describes the scheduling process and the simulation setup and the results are discussed in Sec. VII. Concluding remarks and a brief description of ongoing work are presented in Sec. VIII.

II. Peripheral–Central Vision System

In this paper, the threat aircraft position and attitude are assumed to be obtained from a PCV camera system mounted on a host UAS, as in Fig. 1. This camera system comprises an omnidirectional camera and a PTZ camera with a narrow FOV [13], along with the computer vision algorithms needed to detect and characterize other small UAS that may pose a threat. The omnidirectional camera provides continuous coverage of the environment, but with relatively low and nonuniform resolution. The PTZ camera complements the peripheral vision camera by providing a high resolution image when cued to observe a given threat. The higher resolution PTZ camera image allows classification and attitude estimation [15] of a threat aircraft, while the two cameras together enable stereo ranging of objects within the PTZ camera's FOV [13].

The primary advantage of an airborne small UAS sensing system over a ground-based system is its ability to maneuver to improve detection and characterization performance. Other potential benefits include the ability to pursue or evade threats. In addressing the optimal threat scheduling problem considered here, we make the simplifying assumption that a single host aircraft hovers in one place. However, the overall system capability could be improved by allowing multiple, maneuvering host aircraft to coordinate detection and tracking.

Fig. 1 Peripheral–central vision (PCV) system setup.

Figure 2 illustrates the operation of the sensing system. At the left, the peripheral vision system initially detects a threat in order to cue the PTZ camera for further characterization. To accomplish this, the omnidirectional camera image must first be undistorted, using knowledge of the camera optics, and then stabilized to compensate for camera motion. Optical flow is used to detect threats in the undistorted, stabilized imagery. Once the threat bearing angle is converted from the omnicam frame to the PTZ camera frame, the gimbal adjusts the PTZ camera's attitude to image the threat. The higher resolution image of the central vision camera enables threat classification using a deep neural network (DNN), and the threat attitude can be estimated using a method such as the one presented in [15]. The pair of images together (one from the omnicam and the other from the PTZ camera) enables stereo ranging to determine the threat position, although the accuracy of this estimate depends on the separation (baseline) between the two cameras [13].

For the system described in [13], the Insta360 Air is used as the omnidirectional camera; the two fisheye lenses provide 360° coverage in both azimuth and elevation. An omnicam with greater resolution, such as the all-sky camera mentioned in [10], could decrease threat localization error; the Insta360 Air was selected based on SWaP-C and compatibility with commonly used hardware and software. The PTZ camera system comprises a GigE DFK Z12G445 color zoom camera from The Imaging Source and an HDAir Infinity MR S2 gimbal. The two cameras communicate through ROS, installed on an onboard NVIDIA TX2, which runs all of the system algorithms, such as detection and cueing.

The system architecture described here helps to motivate the problem of detecting and tracking small UAS within a region. Note that the focus is on sensing for counter-UAS applications; the task of removing threats is beyond the scope of this paper.

III. Risk Assignment

The definition of the *risk* posed by a threat may differ depending on the purpose of the sensing system. Defining *threat* and *risk* are essential steps in developing C-UAS systems [1,3,30–32]. For example, if the purpose of the sensing system is just to detect and avoid other aircraft, the minimum time to collision might be an appropriate definition of risk. However, if a sensing system supports counter-UAS to protect a region from overflight by hostile drones, then additional parameters should be considered (e.g., speed, heading, performance capability). Other threat scheduling studies mentioned above [4,16–19] thus have different definitions of risk.

In this paper, the risk posed by a given threat is related to its likelihood of colliding with the host aircraft. The resulting problem formulation emphasizes protecting a point in space, i.e., the host location, rather than a region, but it could easily be extended to the latter case with some simple modifications to the risk measures

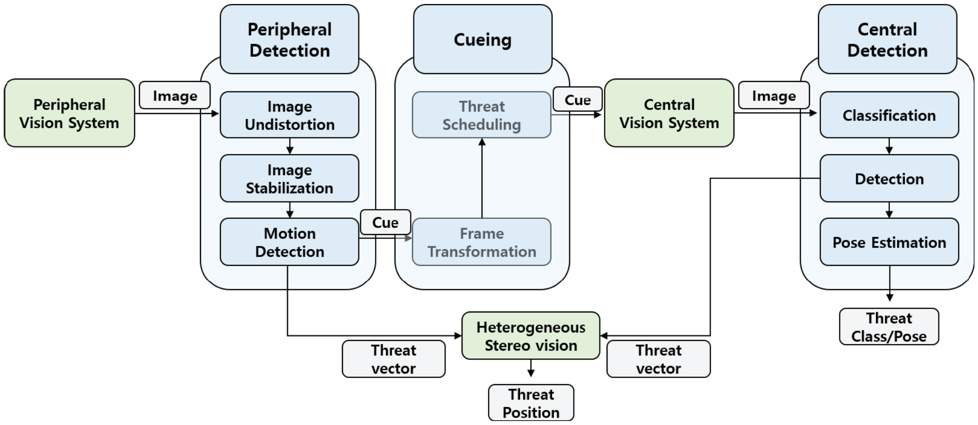


Fig. 2 Peripheral–central vision (PCV) system flow chart.

defined below. To simplify this initial problem formulation, we make the following assumptions:

1) A single host system hovers in place and is tasked to detect and track airborne threats within its sensing range.

2) An algorithm predicts the future path of each detected threat using data provided by the host (e.g., [15,33–38]).

Further, with a focus on counter-UAS applications, we suppose that each threat is a small UAS with corresponding limits on size, speed, and maneuverability.

Given the assumptions above, we define a set of parameters to characterize the risk posed by each threat. Each of these risk parameter values is normalized to a value between 0 and 1, as in [4], and a weighted summation serves as the final measure of risk for the given threat.

A. Risk Parameters

1. *Range to the Threat*

A threat that is closer to the host is considered higher risk. The range to a threat from the host system at time step k is

$$d_k^t = \|\boldsymbol{r}_k^h - \boldsymbol{r}_k^t\| \tag{1}$$

where $\boldsymbol{r}_k^h$ is the host system position and $\boldsymbol{r}_k^t$ is the threat position at time step k. The range is normalized using the maximum detectable range $d_{\max}$ of the sensing system. The value of $d_{\max}$ depends on this system architecture. For the system described in [13], $d_{\max} \approx 100$ m. The normalized risk parameter corresponding to the range to the threat at time step k is

$$\bar{D}_k = \max\left(0, 1 - \frac{d_k^t}{d_{\max}}\right) \tag{2}$$

The parameter is minimum when the threat is at the limits of the detectable range and increases with proximity to the host system. If the focus is on protecting a region, rather than a single point, one could instead compute the distance between the threat and the set defining that region. A spherical “protection region” centered at the host would be a trivial extension; a more general geometry would also be straightforward, though computation time would increase.

2. *Current Approach Speed*

The faster a threat is approaching, the higher the risk that it poses. The threat’s approach speed at time step k can be estimated using a simple finite difference of the distance from the host to the threat over the past two time steps, where the time step Δt corresponds, for example, to the camera frame rate:

$$V_k^t = \frac{1}{\Delta t}(d_k^t - d_{k-1}^t) \tag{3}$$

The estimate (3) is simple to compute, but it is sensitive to error in the distance measure. The estimate could be improved by using a discrete time filter.

Note that the current approach speed V_k^t is positive when the threat is closing on the host. The value is normalized using an estimate of the maximum approach speed ($V_{\max}$) of the threat. Because we have assumed that the host is hovering in place, the value of $V_{\max}$ is simply the maximum speed of the threat; more generally, one would correct for the host’s motion. For a threat that has not been classified, one might choose a sufficiently conservative value for $V_{\max}$. The normalized risk parameter corresponding to the threat’s approach speed at time step k is

$$\bar{V}_k^t = \max\left(0, \frac{V_k^t}{V_{\max}}\right) \tag{4}$$

3. *Predicted Approach Speed*

The predicted path of a threat can be estimated using path prediction algorithms based, for example, on point mass models for threat motion [15,33–39]. The Kalman filter [40–42] and its variants are especially useful for model-based path prediction. Figure 3 shows an aircraft flying away from the host system ($V^t < 0$). However, the predicted path of the aircraft (blue line) implies that the aircraft will change its heading and possibly approach the host system in the future. Even if a threat does not seem risky now, the threat may become riskier if the threat changes its heading. Therefore, the future maneuver of the threat is an important factor to consider for risk value estimation.

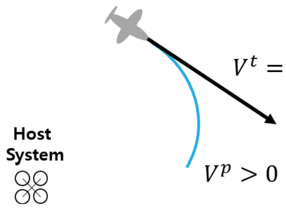


Fig. 3 Current and predicted approach speed.

In [33–39], the aircraft is modeled as a point mass. Its position n time steps in the future can be estimated as

$$\boldsymbol{r}_{k+n}^p = \boldsymbol{r}_k^t + (n\Delta t)\boldsymbol{v}_k^t + \frac{(n\Delta t)^2}{2}\boldsymbol{a}_k^t \tag{5}$$

where $\boldsymbol{r}_k^t$, $\boldsymbol{v}_k^t$, and $\boldsymbol{a}_k^t$ are the position, velocity, and acceleration of the threat aircraft, respectively, at time step k. While the current approach speed V_k^t is computed based on the threat position history over the past few time steps (3), the predicted approach speed at some time n steps in the future is computed at time step k using the predicted position:

$$V_k^p = -\frac{1}{n\Delta t}(d_{k+n}^p - d_k^p) \tag{6}$$

where $d_i^p = \|\boldsymbol{r}_i^h - \boldsymbol{r}_i^p\|$. One may pick a worst-case value of the time step n, using knowledge or assumptions about the speed and agility of a given threat aircraft. A fixed-wing threat, for example, has a limited turn rate for a given speed and this limit can be used to compute the time required to establish a collision course.

Although it is possible to predict a future closing rate for the threat, this prediction relies on a path prediction that will generally be in error. To reduce the likelihood that a high path prediction error will artificially inflate the risk, the predicted approach speed is discounted based on the current path prediction error. In [15], an averaged distance error at time step k between the n-step threat position history and the predicted position history over the same time horizon is computed as follows:

$$e_k^p = \frac{1}{n}\sum_{i=k-n}^{k}\|\boldsymbol{r}_i^t - \boldsymbol{r}_i^p\| \tag{7}$$

See Fig. 4. Using e_k^p to characterize the path prediction error over the recent past and normalizing by a worst-case prediction error ($e_{\max}^p$) gives

$$\bar{e}_k^p = \min\left(\frac{e_k^p}{e_{\max}^p}, 1\right) \tag{8}$$

Considering e^p as a measure of the deviation between the threat position history and the predicted position history, one may choose a conservative value for $e_{\max}^p$ such as the true maximum (absolute) value of e_k^p over the given time history. In this case, $\bar{e}_k^p$ represents a scaling of the prediction error that will fall within the range from 0 to 1, with 1 indicating that the prediction error is so high that the predicted approach speed is unusable as a measure of risk. Scaling the (normalized) predicted approach speed by the factor $1 - \bar{e}_k^p$ gives the following risk parameter:

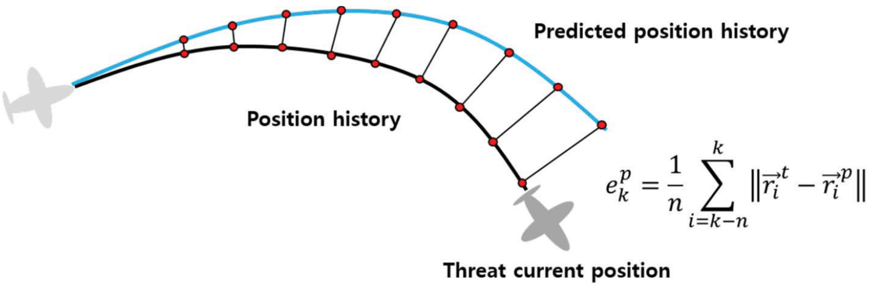


Fig. 4 Computing the position prediction error e^p.

$$\bar{V}_k^p = \max\left(0, \frac{V_k^p}{V_{\max}}(1 - \bar{e}_k^p)\right) \tag{9}$$

4. New Threat Value

If a new threat is observed by the omnidirectional sensor and has not yet been imaged by the directional sensor, this threat may be assigned a high priority in order to obtain the more detailed information required to assess the risk that it poses. Accordingly, we define a parameter c_{nt} called the new threat value. If a new threat is detected, then the parameter value for this threat is $c_{\mathrm{nt}} = 1$; otherwise, $c_{\mathrm{nt}} = 0$.

$$c_{\mathrm{nt}} \in \{0, 1\} \tag{10}$$

B. Total Risk

After estimating the risk parameters defined in Sec. III.A, these parameters are combined as a weighted sum into a single risk value for each known threat at time step k:

$$RV_k = c_{\mathrm{nt}} + (1 - c_{\mathrm{nt}})(W_D\bar{D}_k + W_{V^t}\bar{V}_k^t + W_{V^p}\bar{V}_k^p) \tag{11}$$

Here, the weights are defined such that $W_D + W_{V^t} + W_{V^p} = 1$. Note that $RV = 1$ when $c_{\mathrm{nt}} = 1$, reflecting the immediate prioritization of previously unobserved threats; the addition or removal of threats detected by the omnidirectional sensor is managed by a separate module that is not discussed here. Note that $RV_k \in [0, 1]$.

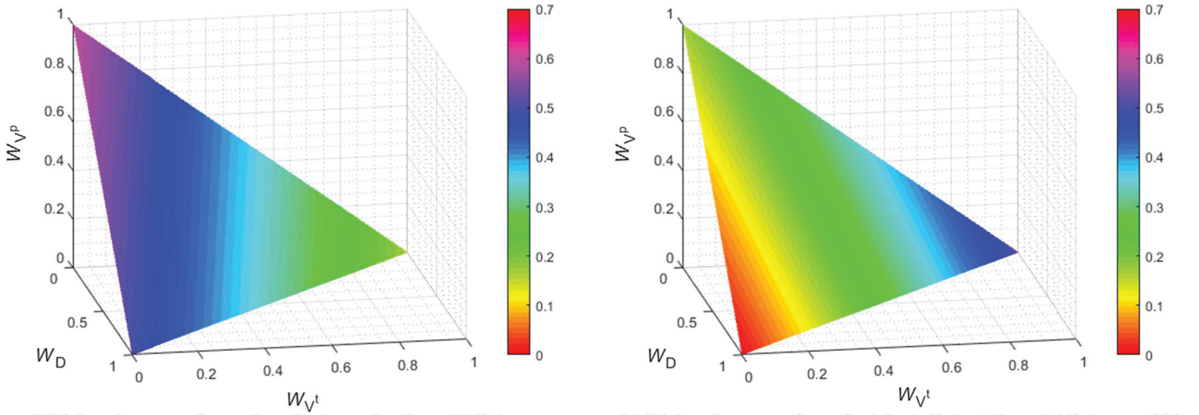
The weight parameters can be determined depending on the user's intention. If one wants to place twice the importance on the current approach speed as other parameters, for example, then one may choose $W_{V^t} = 0.5$ and $W_D = W_{V^p} = 0.25$. Figure 5 shows how the risk value varies with the risk parameter weights for two example sets of risk parameter values for a given threat. Each axis represents a weight, and the risk values are represented by the color map on the triangular portion of the plane $W_D + W_{V^t} + W_{V^p} = 1$. For this example, $d_{\max} = 100$ m, $V_{\max} = 59$ m/s, and $e_{\max}^p = 40$m. Figure 5a corresponds to a threat that is approaching slowly ($\bar{V}^t = 0.2$) from nearby ($\bar{D} = 0.5$) and with a fast predicted approach speed based on an accurately predicted path ($\bar{V}^p = 0.59$). For this threat, the larger W_{V^t} is, the smaller the risk value is because $\bar{V}^t$ is smaller than other parameters. For the example in Fig. 5b, the threat is approaching faster ($\bar{V}^t = 0.5$) but from a more distant location ($\bar{D} = 0$) and with a fast predicted approach speed that is based on an inaccurate path prediction ($\bar{V}^p = 0.17$). Note that $\bar{V}^p$ is much smaller than in the previous example because of the high prediction error, even though V^p is the same for both cases. These examples illustrate how the risk parameter weights determine which parameters will dominate the computed risk value for a given threat. These weights can be modified depending on the threat scheduling scenario.

IV. Modified Traveling Salesman Problem

The threats can be scheduled for the directional sensor based on the risk values computed in the previous section. The directional sensor then reorients in order to observe the threats, starting from the highest priority threat and proceeding to the lowest. As mentioned in Sec. I, both the risk posed by the threats and the need to image the threats expeditiously should be considered in determining the order in which the threats should be imaged by the higher resolution directional sensor.

Figure 6 shows a planar example of a directional sensor maneuver. The triangles depict the FOV of the sensor. In Fig. 6a, threats are prioritized only based on the risk values (T1–T2–T3); the sensor begins with threat 1 and then moves on to threats 2 and 3. However, the observation can be done in a faster way (T1–T3–T2), as shown in Fig. 6b. In this way, threat observation can be conducted in a time-efficient way, so that more threats can be observed in the same time period, and the threat data can be updated more often.

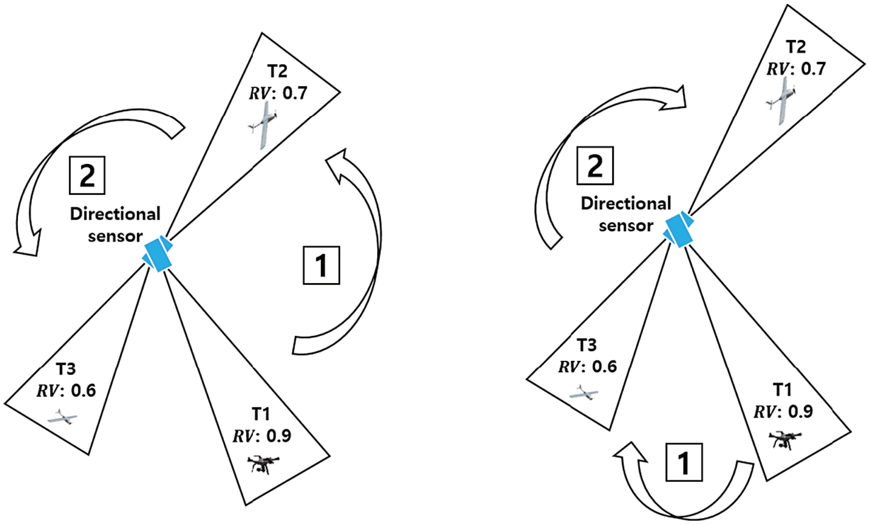
The TSP involves finding the shortest path for a salesman who wants to visit each city in a finite set exactly once. A conventional TSP formulation can be used to determine the fastest way to image all



a) Risk value map for a slow, but nearby threat (distance: 50 m; current approach speed: 10 m/s; predicted approach speed: 40 m/s; prediction error: 5 m)

b) Risk value map for a fast, but distant threat (distance: 100 m; current approach speed: 30 m/s; predicted approach speed: 40 m/s; prediction error: 30 m)

Fig. 5 Risk value versus risk parameter weights for two examples ($d_{\max} = 100$ m, $V_{\max} = 59$ m/s, $e_{\max}^p = 40$ m).



a) Sensor maneuver based only on risk values

b) Sensor maneuver based on risk values and slewing time

Fig. 6 Directional sensor maneuver examples.

threats. In this problem formulation, each threat is treated as a node (or "city"). To determine the optimum tour which visits all n nodes, we first define the cost to slew from threat i to threat j, c_{ij}. Assuming that the sensor moves between threats at a constant rate, and dwells on each threat for the same amount of time, the TSP solution minimizes the total cost of servicing all threats. For the fastest path of the salesman, an integer linear programming problem is formulated as below [43,44]. Here, a binary variable x_{ij} represents whether the camera proceeds from threat i to threat j: if so, then $x_{ij} = 1$; otherwise, $x_{ij} = 0$. Given a subset S of all the nodes, we seek a TSP tour (i.e., values of x_{ij}) that satisfies the following:

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{i\neq j, j=1}^{n} c_{ij} x_{ij} \tag{12}$$

$$\text{s.t.} \quad x_{ij} \in \{0, 1\}, \qquad i, j \in \{1, \ldots, n\} \tag{13}$$

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1, \qquad i \in \{1, \ldots, n\} \tag{14}$$

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1, \qquad j \in \{1, \ldots, n\} \tag{15}$$

$$\sum_{i\in S} \sum_{j\in S} x_{ij} \leq |S| - 1, \qquad \forall\ S \subsetneq \{1, \ldots, n\}, \qquad 2 \leq |S| \leq n - 2 \tag{16}$$

where $|S|$ represents the number of elements (nodal indices) in the subset S. Constraints (14) and (15) ensure that only one path proceeds from each node. Constraint (16) prevents the formation of subtour, i.e., a tour that does not service every threat. For the conventional TSP formulation, the Euclidean distance between nodes i and j defines the cost c_{ij}. In this paper, the TSP formulation is modified to emphasize both the speed with which threats are serviced and the prioritization of higher risk threats. The position of each threat is represented as the line of sight to that threat; thus the angular distance ω_{ij} between threat i and threat j is used for c_{ij} rather than the Euclidean distance; see Fig. 7.

$$\omega_{ij} = \frac{2\sin^{-1}\sqrt{\sin^2(\Delta\theta_{ij}/2) + \cos\theta_i \cos\theta_j \sin^2(\Delta\psi_{ij}/2)}}{\pi} \tag{17}$$

where $\Delta\theta_{ij}$ and $\Delta\psi_{ij}$ are the absolute difference of elevation and azimuth angles between threat i and j. Note that the angular distance is normalized with π as in Sec. III because the maximum angular distance is π. If multiple threats are located with a dense distribution and more than two threats are within the directional sensor's FOV, the threats might be observed simultaneously. In this case, a clustering method with a cluster size constraint determined by the sensor FOV can be used. In this case, the algorithm could proceed with each cluster treated as a single threat, located at the centroid of the cluster, and taking the maximum value of RV of the threats contained in the cluster. This strategy would require that implementation issues (e.g., camera focus) be resolved so that all threats within the cluster can be characterized from the resulting image(s).

In modifying the TSP to account for both observation time and threat risk, RV should be included in c_{ij} along with the slewing time, as characterized by ω_{ij}. As shown in Fig. 6b, if RV is similar for threats 2 and 3, then threat 3 can be observed along the way from threat 1 to threat 2. On the other hand, if RV for threat 3 is low (e.g., 0.1), threat 3 should be ignored because it is much more urgent to image threat 2. Accordingly, we define

$$c_{ij} = \omega_{ij}(1 - W_{RV}) + k_j W_{RV} RV_j \tag{18}$$

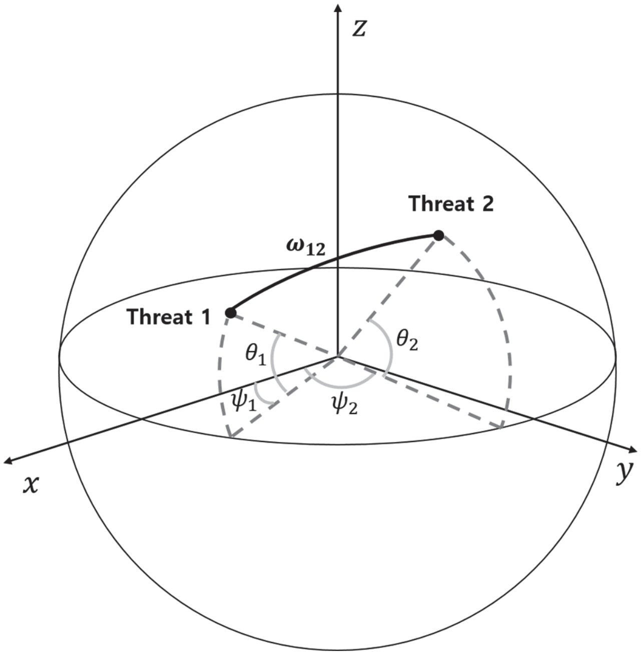
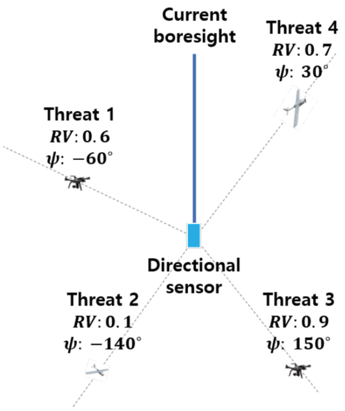


Fig. 7 Angular distance between two threats.



k_j (Threat Order)	Current Position	1	2	3	4
j (Threat #)		3	4	1	2
RV_j		0.9	0.7	0.6	0.1
ω_{ij}		0.833	0.667	0.500	0.444
c_{ij}		**0.866**	**1.033**	**1.150**	**0.422**

$W_{RV} = 0.5$, Total cost: 3.472

k_j (Threat Order)	Current Position	1	2	3	4
j (Threat #)		4	3	2	1
RV_j		0.7	0.9	0.1	0.6
ω_{ij}		0.167	0.667	0.389	0.444
c_{ij}		**0.433**	**1.233**	**1.094**	**0.422**

$W_{RV} = 0.5$, Total cost: 3.433

k_j (Threat Order)	Current Position	1	2	3	4
j (Threat #)		4	3	1	2
RV_j		0.7	0.9	0.6	0.1
ω_{ij}		0.167	0.667	0.833	0.444
c_{ij}		**0.433**	**1.233**	**1.316**	**0.422**

$W_{RV} = 0.5$, Total cost: 3.405

Fig. 8 Candidate tours for a modified TSP example.

where the integer $k_j \in \{1, \ldots, n\}$ represents the prioritization of threat j. A smaller value indicates higher priority; for example, if threat 3 has the highest priority, then $k_3 = 1$. The parameter $W_{RV} \in [0, 1]$ balances the importance of imaging threats that pose a high risk against the need to image all of the threats as quickly as possible. The greater the value of W_{RV}, the more heavily the risk value RV is weighted in determining the optimum tour.

Given values for the risk parameters associated with each threat and the user-defined weights, the TSP solver computes the cost of each possible tour through the threats and selects the one for which the total cost is minimum. Figure 8 shows a planar example of the modified TSP. There are four threats, each with a unique azimuthal position ψ. The thick blue line represents the current boresight of the directional sensor, i.e., the starting orientation for the tour of the four threats. (Although a planar example is overly simplistic, it serves as a helpful illustration.) The tables in Fig. 8 show the cost for 3 of the 24 possible tours. The table at the top shows the cost computation when the threat order is T3–T4–T1–T2, which services the threats in descending order of risk. To compute the cost c_{c3} to reorient the directional sensor from its initial orientation to threat T3, the algorithm sets $k_3 = 1$ with $\omega_{c3} = 0.833 = 30°/180°$. In this case,

$$c_{c3} = 0.833 \times (1 - 0.5) + 1 \times 0.5 \times 0.9 = 0.866$$

In a similar way, the cost of each leg of the tour is computed up to

$$c_{12} = 0.444 \times (1 - 0.5) + 4 \times 0.5 \times 0.1 = 0.422$$

The total cost of the tour is given below the top-most table: 3.472. If the threat order is changed to T4–T3–T2–T1, as shown in the table in the middle, the total cost decreases to 3.433. The reason is that the risk values of T4 and T3 are similar, so observing T4 along the way to T3 is more time-efficient. But because T2 is not an urgent threat ($RV = 0.1$), imaging this threat could be postponed, given that T1 poses a higher risk ($RV = 0.6$). If we change the order as in the table at the bottom (T4–T3–T1–T2), the cost decreases further to 3.405. For these examples, $W_{RV} = 0.5$. If one wishes to emphasize observation time efficiency more heavily than risk, one might choose $W_{RV} = 0.3$. In this case, the total cost of the tour in the middle table becomes 2.7269, whereas the cost of the tour at the bottom becomes 2.8877. In this case, the optimization algorithm indicates that T2 should be observed earlier than T1, despite the lower risk that T2 poses. Thus, the parameter W_{RV} adjusts the balance of priority between risk and time efficiency. In the next section, the detailed threat scheduling algorithm design is described using the modified TSP solution and the risk value estimation.

V. Threat Scheduling Algorithm Design

Based on the risk value estimation and the modified TSP solution introduced in previous sections, the threat scheduling algorithm can now be presented. The algorithm runs in a continuous loop that resembles the observe-orient-decide-act (OODA) cycle attributed by Schechtman [45] to Col. J. R. Boyd. Figure 9 shows the threat scheduling flow chart for the PCV system. The peripheral vision camera provides continuous visual coverage, providing continuous bearing data (i.e., line-of-sight angles) for all detected threats. In the initial observation cycle, detailed threat data such as position, velocity, and the predicted path are not available because the central vision camera has not yet imaged the threats. To initialize the process, the central vision camera scans all threats as quickly as possible, following the tour provided by the conventional TSP solution (with $W_{RV} = 0$).

Once the initial observation cycle is over, risk parameters can be computed for each threat. Although all threats have now been imaged, the next tour has not been optimized. All of these threats are declared “unobserved” and added to the “unobserved threat list.” Next, the modified TSP is solved for all threats in the unobserved threat list. Once the threat schedule is determined, the central vision camera starts to slew to the first threat in the prioritized list. Once this threat is detected by the central vision camera, the state (position, velocity, etc.) is estimated, the risk parameter values are updated, and the threat is moved to the “observed threat list” from the unobserved threat list. In practice, the central vision camera may be unable to acquire the threat for some reason, so the central vision camera is commanded to move to the next threat after an observation time limit t_o. Each time a threat is observed by the central vision camera, the modified TSP is solved for the unobserved threats again. Thus, any new threat that appears in the omnidirectional camera image will be immediately serviced by the directional camera. Once every threat has been observed, so that the unobserved threat list is empty, all threats contained in the observed threat list are moved to the unobserved threat list. This process runs as a continuous loop. In the following section, the threat scheduling algorithm is implemented in a simulation to assess the performance.

VI. Simulation Setup

In this section, a simulation is conducted to assess the performance of the suggested threat scheduling algorithm. For simplicity, we assume that the altitudes of all threats and the PCV system are the same so that we may consider motion in a horizontal plane. The threat scheduling process using the PCV system (Fig. 9) is depicted in the simulation, and the simulation parameters are shown in Table 1.

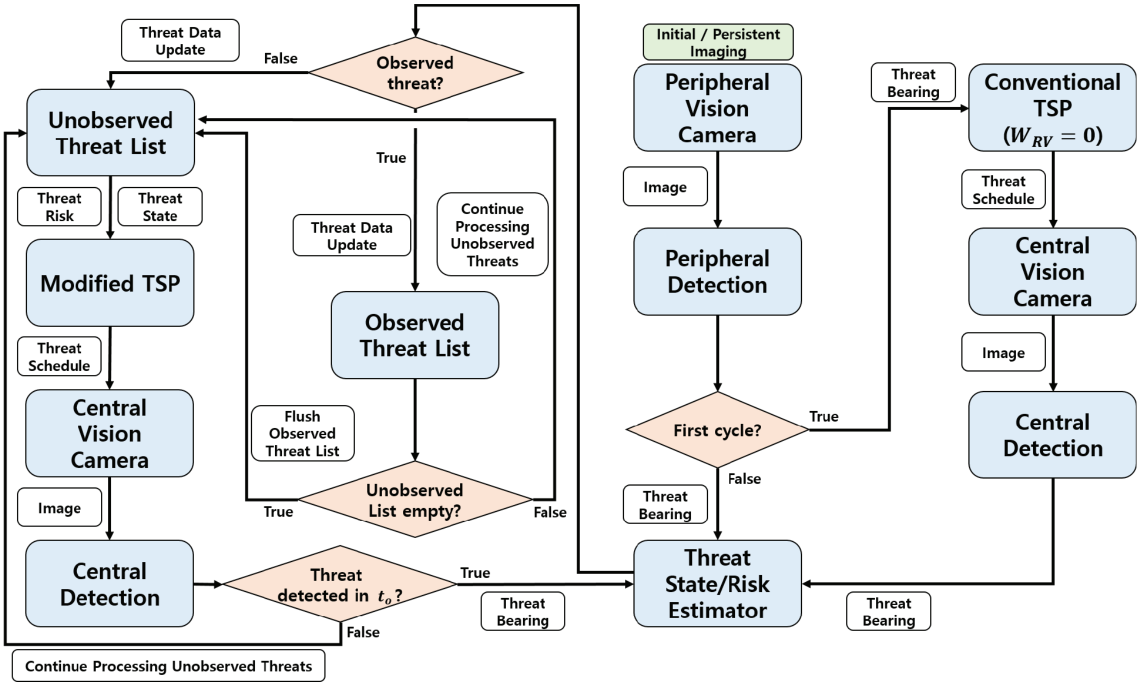


Fig. 9 Threat scheduling flow chart for the PCV system.

The maximum detectable range of the PCV system is $d_{\max} = 100$ m. All threats are assumed to be small UAS, and the typical racing drone speed of 59 m/s [46] is taken as $V_{\max}$. We chose 40 m for $e^p_{\max}$ based on actual small UAS flight data [47]. To apply the same weight for $\bar{D}$ and $\bar{V}^t$ for threat scheduling, W_D and W_{V^t} are defined as 0.4, and we used 0.2 for W_{V^p} to emphasize $\bar{D}$ and $\bar{V}^t$ more than $\bar{V}^p$.

Figure 10 shows an example of the simulation. The simulation first generates threats with random data (position, speed, and attitude) with the simulation parameters shown in Table 1 as in Fig. 10a, and threats do the steady turning flight and randomly change their roll angles during the simulation. In the figure, the PCV system is located at the origin and the threats have been detected by the peripheral vision camera. The only data obtained from the peripheral vision camera are the line of sight to each threat, so the risk parameters for the threats are not available in the first cycle. Based on the bearing data, the conventional TSP solution ($W_{RV} = 0$) is adopted to quickly image all the threats; this initial threat schedule is 2-3-5-1-4. The boresight of the central vision camera (depicted by a thick blue line) starts to move based on the schedule. (We assume simple dynamics for the camera gimbal, as one would obtain using a proportional-derivative servo controller.) Once a threat is observed (Fig. 10b), the position, velocity, and predicted path (red lines) of the threat are estimated. Note that the position of each threat is updated to the estimated value in Fig. 10b. Having quickly acquired initial data concerning the threats, the risk parameter values are computed using the approach presented in Sec. III and a new threat schedule (4-1-5-2-3) based on the modified TSP solution (with $W_{RV} = 0.5$) is then determined; see Fig. 10c. In this example, threat #4 has the highest risk value (it is approaching the PCV system from nearby), and it is designated by the TSP solver as the highest priority threat. Threat #1 is designated as the second priority, because it is quite close to threat #4. After observing threat #1, the directional sensor is steered to threat #5, which has the second-highest risk value (0.27). Threat #2 follows because threat #3 has a zero risk value.

Table 1 Simulation parameters

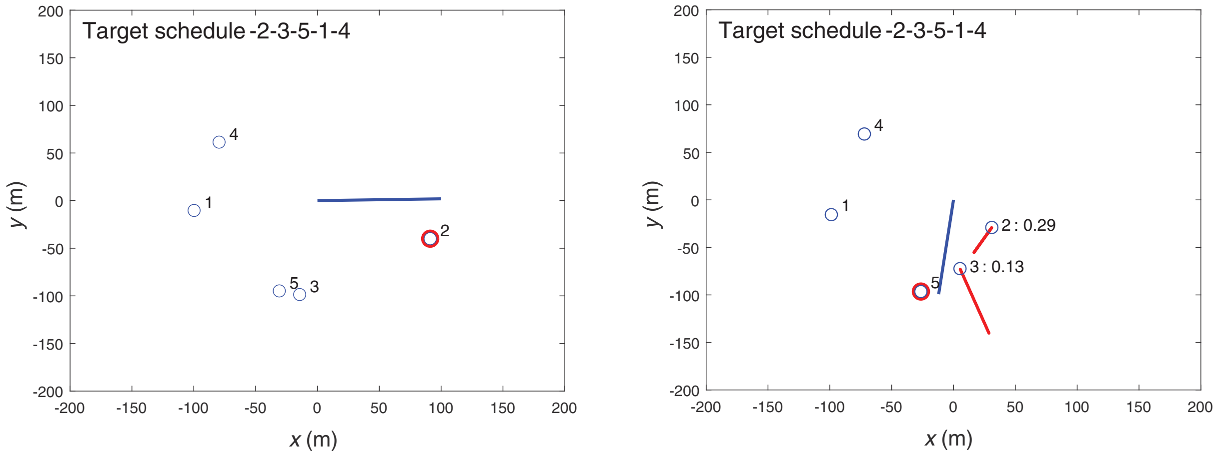
Parameter	Value
Simulation time	5 min
Threat initial range	80–100 m
Threat roll angle	0–10°
Threat heading angle	0–360°
Gimbal maximum angular speed	40°/s
Number of threats	8
$d_{\max}$	100 m
$V_{\max}$	59 m/s
$e^p_{\max}$	40 m
W_D, W_{V^t}	0.4
W_{V^p}	0.2

VII. Discussion of Results

The weight parameter W_{RV} determines the balance between risk and time-efficiency in prioritizing observations. Figure 11 shows the modified TSP solutions for different values of W_{RV} in the threat scheduling simulation. For the left figure $W_{RV} = 0$ and the threat schedule (2-4-1-5-3) minimizes the scanning time. This approach enables fast threat scanning but gives no special priority to threats that may pose a high risk. For the right figure $W_{RV} = 1$ and the schedule (5-2-4-3-1) considers only the risk posed by each threat, as in other studies [4,16,19], without considering the time required to service the threats.

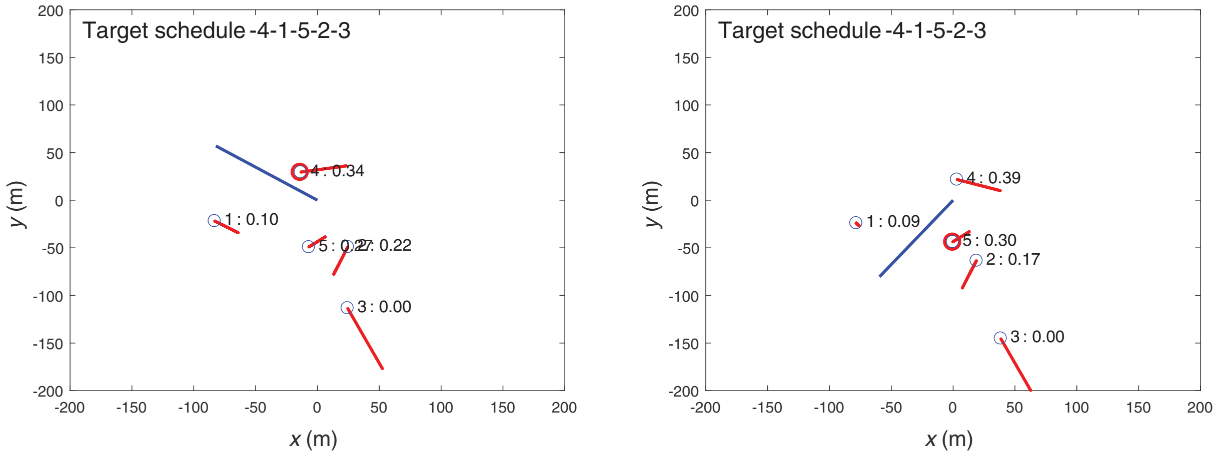
To assess the performance of the modified TSP algorithm for multiple threat scheduling, simulations were conducted using the parameters shown in Table 1. The purpose of these simulations is to compare the performance between purely risk-based scheduling ($W_{RV} = 1$), as in [16–19]; purely time-based scheduling ($W_{RV} = 0$), as in [29]; and the proposed method for balancing these two concerns ($0 < W_{RV} < 1$).

In each of these simulations, eight threats were placed randomly within the scene and they maneuvered through the omnicam FOV over a period of 5 minutes. Two simulations were run at each of 50 values of W_{RV} ranging from 0 to 1. For each simulation, three performance measures were computed: a measure of the risk value prioritization, the total number of threats observed during a simulation, and the differential entropy, which is discussed a bit later. The risk value prioritization was measured by summing the risk values RV for the four *highest priority* threats each time the unobserved list was refreshed and averaging all of these outcomes over the simulation. The resulting value ranges from 0 to 1, with values closer to 1 indicating that threats posing the greatest risk have been prioritized. Because 100 simulations were performed at each value



a) The initial threat schedule is minimum time

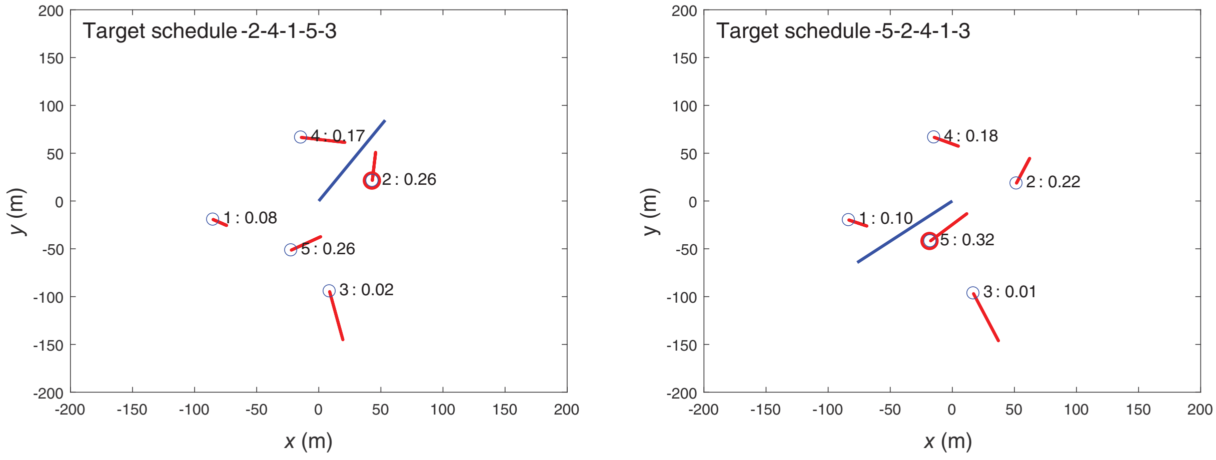
b) A risk value is assigned to each threat, once observed by the central vision camera



c) Once all threats have been observed, a threat schedule may be computed based on weighted risk

d) Risk values are updated in the order indicated by the modified TSP

Fig. 10 Threat scheduling simulation (red asterisk: PCV system; blue circles: threats; thick blue line: boresight of the central vision camera; red circle: next threat to observe; red lines: predicted threat paths; numerical annotations: threat number and risk value).



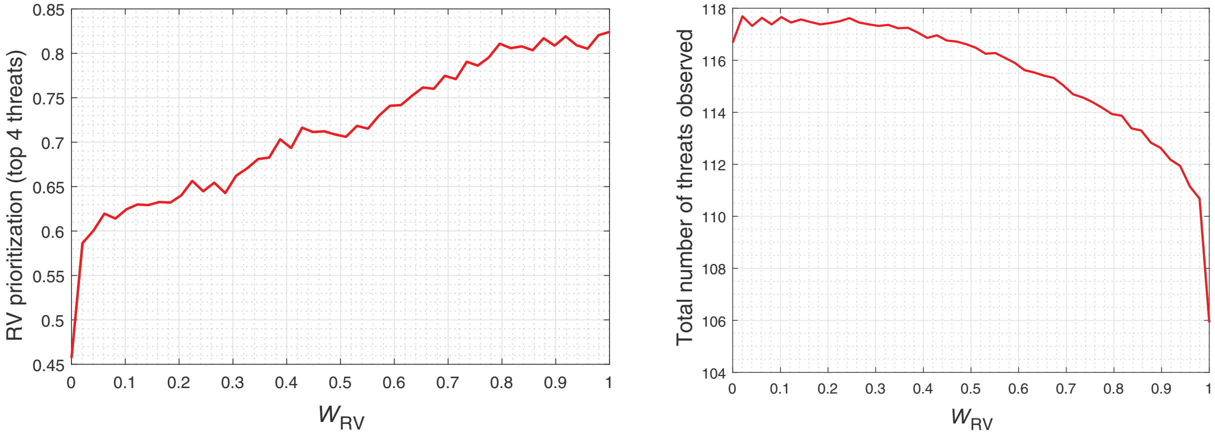
a) Observation time-based threat scheduling ($W_{RV} = 0$)

b) Risk-based threat scheduling ($W_{RV} = 1$)

Fig. 11 Modified TSP solution W_{RV}.

of W_{RV}, 100 values were obtained for each of these three measures. These 100 values were then averaged. The results for risk value prioritization and the number of threats observed are shown in Fig. 12. (The trends in these figures were similar to those obtained using 30 and 70 simulations, so 100 simulations are sufficient to understand the qualitative variation in system performance.)

Figure 12a shows the summed risk value for the four threats that are assigned the highest priority by the algorithm. A higher value indicates that the algorithm has prioritized threats that pose a greater risk. When $W_{RV} = 0$, only the observation time is considered for scheduling, so summed risk value of the top four threats is lowest. After an initial jump, the summed risk value increases almost linearly



a) Summed risk values for the top 4 threats vs W_{RV} **b) Number of threats observed vs W_{RV}**

Fig. 12 Risk prioritization and observation speed versus W_{RV}.

with W_{RV} as W_{RV} increases from 0 toward 1. Figure 12 shows the total number of threats observed by the central vision camera during the 5-minute interval. In contrast with Fig. 12a, the total number of threats observed is largest when W_{RV} is close to 0 and decreases almost linearly with decreasing values of W_{RV}, with a sharp drop-off at $W_{RV} = 1$. For low values of W_{RV}, each of the 8 threats is imaged between 14 and 15 times, on average, over the 5-minute interval. For this particular set of simulations, choosing $W_{RV} = 0.8$ rather than $W_{RV} = 1$ would ensure that each threat is observed one additional time, on average, over the 5-minute period with a negligible change in the risk measure.

If the central vision camera can observe more threats, the threat data can be updated more often, so the uncertainty in estimates of threat motion can be decreased. To see the trends of the 2×2 covariance matrix Σ for the estimated threat position, we computed the differential entropy

$$H = \frac{1}{2}\log[(2\pi e)^2|\Sigma|] \tag{19}$$

where $|\Sigma|$ represents the determinant. High differential entropy represents large uncertainty. Figure 13 shows the mean differential entropy of all the estimated threat positions versus W_{RV}. The differential entropy increases with W_{RV} and rises sharply near $W_{RV} = 1$, illustrating how the tradeoff between faster (more frequent) imaging and risk-prioritized imaging affects overall estimate uncertainty. Although the y-axis scale in Fig. 13 seems small, the results indicate that the value of $|\Sigma|$ decreases by 10% as W_{RV} drops from 1 to about 0.6.

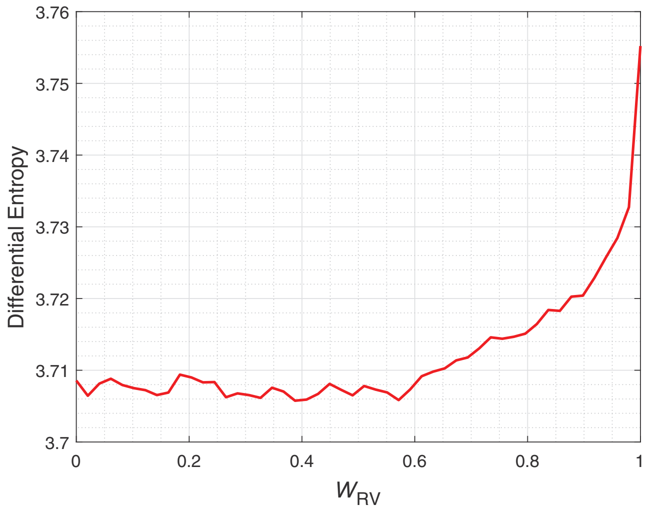


Fig. 13 Differential entropy versus W_{RV}.

As shown in these figures, low values of W_{RV} help the central vision camera to move more efficiently so that the central vision camera captures more threats in the same period of time, and the measurement uncertainty of threat data decreases. On the other hand, high values of W_{RV} prioritize high-risk threats to be observed sooner, at some cost to the threat imaging efficiency. Simulation results for the proposed algorithm suggest that relaxing the priority on risk, e.g., by choosing W_{RV} between 0.6 and 0.8, can provide better situation awareness. By servicing threats more quickly, one may obtain more accurate overall assessment of the risk that they pose.

VIII. Conclusions

In this paper, a novel multiple-threat scheduling algorithm using a modified TSP solution for omnidirectional–directional sensor systems is presented. Although various threat scheduling algorithms have been suggested in previous studies, the tradeoff between the risk posed by threats and the speed at which threats are imaged has not been considered. A mobile sensing system, such as the PCV system mentioned here, has a limited range and a limited response speed, so reacting to threats quickly and judiciously is important. To efficiently schedule threats detected by the omnidirectional sensor, a modified TSP algorithm is introduced, and the suggested algorithm is implemented in simulations. The simulation results show that the modified TSP algorithm is capable of prioritizing the threats by considering both the time efficiency and the risk values, which existing threat scheduling algorithms do not do. The results also show that the measurement uncertainty of threat data can be decreased by updating the threat data more often. One concern is the algorithm's computational complexity. Although a clustering approach can be used in a dense threat environment to decrease the number of threats and speed up the algorithm, if the number of threat clusters is large (of order 10), the computation time would increase unmanageably because the TSP computation time increases exponentially with the number of nodes. Thus, the proposed strategy is most appropriate for scenarios where the threat count remains low. Ongoing efforts aim to relax some of the assumptions mentioned in Sec. III, such as the assumption that there is only a single host and this host hovers in one place. It is expected that these extensions would expand the system's range and accuracy as well as helping to manage larger numbers of threats.

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] Michel, A. H., *Counter-Drone Systems*, 2nd ed., Center for the Study of the Drone at Bard College, Annandale-on-Hudson, NY, 2019, pp. 1–13, https://dronecenter.bard.edu/files/2019/12/CSD-CUAS-2nd-Edition-Web.pdf.

[2] Anonymous, "Administrator's Fact Book," Federal Aviation Administration TR SAND2015-6365 606150, Albuquerque, NM, 2019, https://www.faa.gov/news/media/2019_Administrators_Fact_Book.pdf.

[3] Birch, G. C., Griffin, J. C., and Erdman, M. K., "UAS Detection, Classification, and Neutralization: Market Survey 2015," Sandia National Lab., 2015.
https://doi.org/10.2172/1222445

[4] Khosla, D., Huber, D. J., and Chen, Y., "Automated Scheduling of Radar-Cued Camera System for Optimizing Visual Inspection and Detection of Radar Targets," *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, IEEE, New York, April 2017, pp. 1–5.
https://doi.org/10.1109/THS.2017.7943476

[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] 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

[8] 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, 2010, pp. 646–651.
https://doi.org/10.1109/IROS.2010.5652254

[9] 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, 2011, pp. 340–343.

[10] Siewert, S. B., Andalibi, M., Bruder, S., and Rizor, S., "Slew-to-Cue Electro-Optical and Infrared Sensor Network for Small UAS Detection, Tracking and Identification," *AIAA SciTech 2019*, AIAA Paper 2019-2264, Jan. 2019.
https://doi.org/10.2514/6.2019-2264

[11] Hammer, M., Borgmann, B., Hebel, M., and Arens, M., "UAV Detection, Tracking, and Classification by Sensor Fusion of a 360° Lidar System and an Alignable Classification Sensor," *SPIE Defense + Commercial Sensing*, SPIE, Bellingham, WA, May 2019.
https://doi.org/10.1117/12.2518427

[12] Hammer, M., Hebel, M., Laurenzis, M., and Arens, M., "Lidar-Based Detection and Tracking of Small UAVs," SPIE Security + Defence, Oct. 2018.
https://doi.org/10.1117/12.2325702

[13] 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

[14] Zhang, C., and Hwang, I., "Multi-Target Identity Management with Decentralized Optimal Sensor Scheduling," *European Journal of Control*, Vol. 56, Nov. 2020, pp. 10–37.
https://doi.org/10.1016/j.ejcon.2020.01.004

[15] 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

[16] 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

[17] 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.

[18] 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

[19] Ding, Z., and Moo, P., "Benefits of Target Prioritization for Phased Array Radar Resource Management," *2017 18th International Radar Symposium (IRS)*, IEEE, New York, June 2017, pp. 1–7.
https://doi.org/10.23919/IRS.2017.8008153

[20] Marques, T., Lukic, L., and Gaspar, J., "Observation Functions in an Information Theoretic Approach for Scheduling Pan-Tilt-Zoom Cameras in Multi-Target Tracking Applications," *Robot 2015: Second Iberian Robotics Conference*, Springer, Berlin, Nov. 2015, pp. 503–515.
https://doi.org/10.1007/978-3-319-27149-1_39

[21] Sommerlade, E., and Reid, I., "Probabilistic Surveillance with Multiple Active Cameras," *2010 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, New York, May 2010, pp. 440–445.
https://doi.org/10.1109/ROBOT.2010.5509736

[22] Sommerlade, E., and Reid, I., "Cooperative Surveillance of Multiple Targets Using Mutual Information," *Workshop on Multi-Camera and Multi-Modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*, Springer, Berlin, Oct. 2008.

[23] Zhang, C., and Hwang, I., "Decentralized Multi-Sensor Scheduling for Multi-Target Tracking and Identity Management," *2019 18th European Control Conference (ECC)*, IEEE, New York, June 2019, pp. 1804–1809.
https://doi.org/10.23919/ECC.2019.8796293

[24] Siewert, S. B., Andalibi, M., Bruder, S., and Buchholz, J. M., "Comparison of RADAR, Passive Optical with Acoustic, and Fused Multi-Modal Active and Passive Sensing for UAS Traffic Management Compliance and Urban Air Mobility Safety," *AIAA SciTech 2020*, AIAA Paper 2020-1456, Jan. 2020.
https://doi.org/10.2514/6.2020-1456

[25] Loy, G., Fletcher, L., Apostoloff, N., and Zelinsky, A., "An Adaptive Fusion Architecture for Target Tracking," *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, IEEE, New York, May 2002.
https://doi.org/10.1109/AFGR.2002.1004164

[26] Poitevin, P., Pelletier, M., and Lamontagne, P., "Challenges in Detecting UAS with Radar," *2017 International Carnahan Conference on Security Technology (ICCST)*, IEEE, New York, Oct. 2017.
https://doi.org/10.1109/CCST.2017.8167852

[27] Wellig, P., Speirs, P., Schuepbach, C., Oechslin, R., Renker, M., Boeniger, U., and Pratisto, H., "Radar Systems and Challenges for C-UAV," *2018 19th International Radar Symposium (IRS)*, IEEE, New York, June 2018.
https://doi.org/10.23919/IRS.2018.8448071

[28] Neves, J. C., and Proença, H., "Dynamic Camera Scheduling for Visual Surveillance in Crowded Scenes Using Markov Random Fields," *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, New York, Aug. 2015, pp. 1–6.
https://doi.org/10.1109/AVSS.2015.7301790

[29] Del Bimbo, A., and Pernici, F., "Towards On-Line Saccade Planning for High-Resolution Image Sensing," *Pattern Recognition Letters*, Vol. 27, No. 15, 2006, pp. 1826–1834.
https://doi.org/10.1016/j.patrec.2006.02.014.

[30] Michel, A. H., *Counter-Drone Systems*, Center for the Study of the Drone at Bard College, Annandale-on-Hudson, NY, 2018, pp. 1–10, https://dronecenter.bard.edu/files/2018/02/CSD-Counter-Drone-Systems-Report.pdf.

[31] Tyurin, V., Martyniuk, O., Mirnenko, V., Open'ko, P., and Korenivska, I., "General Approach to Counter Unmanned Aerial Vehicles," *2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, IEEE, New York, Oct. 2019, pp. 75–78.

[32] Herrera, G. J., Dechant, J. A., Green, E. K., and Klein, E. A., "Technology Trends in Small Unmanned Aircraft Systems (sUAS) and Counter-UAS: A Five Year Outlook (No. IDA-P-8823, H-17-000624)," Inst. for Defense Analyses Alexandria, 2017.

[33] Slattery, R., and Zhao, Y., "Trajectory Synthesis for Air Traffic Automation," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 2, 1997, pp. 232–238.
https://doi.org/10.2514/2.4056

[34] Vilardaga, S., and Prats, X., "Mass estimation for an Adaptive Trajectory Predictor Using Optimal Control," *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems*, IEEE, New York, 2015, pp. 75–84.
https://doi.org/10.1145/1235

[35] Glover, W., and Lygeros, J., "A Multi-Aircraft Model for Conflict Detection and Resolution Algorithm Evaluation," HYBRIDGE TR WP1, Deliverable D1.3, Version 1.3., 2003.

[36] Ayhan, S., and Samet, H., “Aircraft Trajectory Prediction made Easy with Predictive Analytics,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, Aug. 2016, pp. 21–30.
https://doi.org/10.1145/2939672.2939694

[37] Lymperopoulos, I., Lygeros, J., and Lecchini, A., “Model Based Aircraft Trajectory Prediction During Takeoff,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2006-6098, Aug. 2006.
https://doi.org/10.2514/6.2006-6098

[38] Zhang, J., Jie Liu, R. H., and Zhu, H., “Online Four Dimensional Trajectory Prediction Method Based on Aircraft Intent Updating,” *Aerospace Science and Technology*, Vol. 77, June 2018, pp. 774–787.
https://doi.org/10.1016/j.ast.2018.03.037

[39] Kang, C., and Woolsey, C. A., “Model-Based Path Prediction for Fixed-Wing Unmanned Aircraft Using Pose Estimates,” *Aerospace Science and Technology*, Vol. 105, Oct. 2020, Paper 106030.
https://doi.org/10.1016/j.ast.2020.106030

[40] Luo, C., McClean, S. I., Parr, G., Teacy, L., and De Nardi, R., “UAV Position Estimation and Collision Avoidance Using the Extended Kalman Filter,” *IEEE Transactions on Vehicular Technology*, Vol. 62, No. 6, 2013, pp. 2749–2762.
https://doi.org/10.1109/TVT.2013.2243480

[41] Prevost, C. G., Desbiens, A., and Gagnon, E., “Extended Kalman Filter for State Estimation and Trajectory Prediction of a Moving Object Detected by an Unmanned Aerial Vehicle,” *2007 American Control Conference*, IEEE, New York, July 2007, pp. 1805–1810.
https://doi.org/10.1109/ACC.2007.4282823

[42] Watanabe, Y., Calise, A., and Johnson, E., “Vision-Based Obstacle Avoidance for UAVs,” *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA Paper 2007-6829, Aug. 2007.
https://doi.org/10.2514/6.2007-6829

[43] Dantzig, G., Fulkerson, R., and Johnson, S., “Solution of a Large-Scale Traveling-Salesman Problem,” *Journal of the Operations Research Society of America*, Vol. 2, No. 4, 1954, pp. 393–410.
https://doi.org/10.1287/opre.2.4.393

[44] Laporte, G., “The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms,” *European Journal of Operational Research*, Vol. 59, No. 2, 1992, pp. 231–247.
https://doi.org/10.1016/0377-2217(92)90138-Y

[45] Schechtman, G. M., “Manipulating the OODA Loop: The Overlooked Role of Information Resource Management in Information Warfare,” Master’s Thesis, Air Force Inst. of Technology, Wright-Patterson AFB, OH, Dec. 1996.

[46] Sewart, T., Partridge, L., Cornwell, D., Arcas, D. A., and Wilcox, C., “Quick Quadcopters: Top Speed of a Racing Drone,” *Physics Special Topics*, Vol. 18, No. 1, 2019, p. P1-6.
https://doi.org/10.1016/j.patrec.2006.02.014

[47] McClelland, H. G., Kang, C., Woolsey, C. A., Roberts, A. K., Buck, D., Cheney, T., and Warnick, K., “Small Aircraft Flight Encounters Database for UAS Sense and Avoid,” *AIAA SciTech*, AIAA Paper 2017-1152, Jan. 2017.
https://doi.org/10.2514/6.2017-1152

D. Casbeer
Associate Editor