AIAA AVIATION Forum
August 2-6, 2021, VIRTUAL EVENT
AIAA AVIATION 2021 FORUM

10.2514/6.2021-2393

Check for updates

Optimal Placement Algorithm for Multiple Heterogeneous Stereo Vision Systems

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

An unmanned aircraft system (UAS) can use a heterogeneous stereo vision system to detect and localize other aircraft which may pose a threat. The error in the triangulated location may be large, however, because of the short baseline between the two cameras mounted on the host aircraft. One way to address this issue is to use multiple host aircraft, each with a heterogeneous stereo vision system, in order to increase the stereo vision baseline. In this letter, we suggest a heterogeneous stereo vision optimal placement (HSOP) algorithm that solves a mixed-integer nonlinear programming problem to determine optimal positions for multiple airborne heterogeneous stereo vision systems in order to minimize the localization error for multiple threats. After introducing camera models for the "peripheral-central vision system," we formulate an optimization problem that supports counter-UAS applications. To assess the algorithm's performance, the simulated localization error of the HSOP algorithm is compared with that of another placement algorithm that considers only threat coverage. The results indicate that the HSOP algorithm effectively reduces the threat localization error.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

I. Introduction

A wide variety of beneficial uses for unmanned aircraft systems (UAS) have been demonstrated, but interest in counter-UAS (C-UAS) systems has also been growing, in part because of the costly "asymmetric threat" drones have demonstrated in incidents like the 2019 shutdown of London's Gatwick airport [1]. C-UAS technologies use a variety of sensors such as radar, lidar, cameras, acoustic sensors, and RF sensors. Among these sensors, cameras are especially common because of their low size, weight, power and cost (SWaP-C) and the emergence of very effective computer vision algorithms. Among 323 C-UAS products in the world that are capable of detecting threat aircraft, 113 systems use cameras for detection [2]. For similar reasons, cameras are also common sensors for small UAS (sUAS).

Reference [3] describes development of a "peripheral-central vision (PCV) system" to detect, localize and classify airborne threats using heterogeneous stereo vision. This system uses two types of low-cost cameras to characterize a threat: a wide field-of-view (FOV) "peripheral vision" camera and a narrow FOV "central vision" camera capable of pan-tilt (PT) operation. The peripheral vision camera (e.g., an omnidirectional camera) offers continuous visual coverage of the environment for threat detection, although with relatively low and non-uniform resolution. The central vision camera complements the peripheral vision camera by providing a high-resolution image when cued to observe a threat. The pair of cameras affords the opportunity to use stereo vision for estimating the 3D position of the threat. However, the PCV system is mounted on a single host aircraft, so the camera baseline (the distance between the two cameras) is limited. A short camera baseline generates a large localization error for distant threats. One way to address this issue is to add more host aircraft, each with its own PCV system, in order to extend the effective camera baseline. While a longer baseline can decrease localization error, this is not guaranteed. If the encounter geometry is degenerate, for example, the localization error may still be large. It is therefore of interest to determine optimal positions for additional host aircraft to minimize threat localization error for C-UAS applications.

The focus of this note is a heterogeneous stereo-vision optimal placement (HSOP) algorithm which minimizes the error incurred when localizing a threat aircraft using multiple mobile PCV systems. We begin by introducing models for the peripheral and central vision cameras in a PCV system. Each type of camera has a different FOV and detectable range. The peripheral vision camera has low image resolution that is nonuniform over its FOV so the localization error using this camera is comparatively larger than that obtained using a central vision camera. Because the peripheral

*Graduate Research Assistant, Department of Aerospace and Ocean Engineering
†Professor, Department of Aerospace and Ocean Engineering, AIAA Associate Fellow

Copyright © 2021 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

vision camera has a large FOV, however, it can image multiple threat aircraft simultaneously. Conversely, the central vision camera can provide better localization accuracy, but the narrow FOV makes it less likely to image multiple threats at once. The HSOP algorithm determines the minimum number of PCV-equipped aircraft required to localize a given number of threat aircraft as well as the optimal locations for those host aircraft and the specific camera (peripheral or central) that each should employ to minimize the cumulative localization error. The algorithm inputs are the number of threats and a preliminary estimate of their position and velocity. It is assumed that sufficiently many host aircraft are available to accurately localize all threats.

Currently, the HSOP algorithm allocates exactly two cameras to each threat, though it is possible to further reduce localization error using additional camera imagery. The current formulation also weights each threat equally in the cumulative optimization process. One could easily modify the algorithm to place higher priority on threats that pose a greater risk based on some appropriate metric. While we have assumed each host includes a PCV system, the HSOP algorithm can be used for optimal placement of other mobile, vision-based sensing systems.

The note is organized as follows. Section II reviews work related to this investigation. Section III describes the camera models used for triangulation-based localization. Section IV introduces the HSOP algorithm. Analysis of the algorithm's performance is detailed in Section V. Concluding remarks and a brief description of ongoing work are presented in Section VI.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

II. Related work

Two aircraft, each equipped with a PCV system, can localize a threat using triangulation, provided the threat is viewable by both aircraft. Here, we review prior work concerning optimal placement of mobile sensors such as camera-equipped drones.

The optimal drone placement problem has enjoyed considerable attention within the larger context of wireless sensor networks (WSNs). Many of these studies have addressed ground target coverage problems using multiple camera-mounted drones. Various approaches include the geometric algorithm [4], the brainstorm algorithm [5], the moth search algorithm [6], the bare bones fireworks algorithm [7], and so on. These camera-mounted drone placement algorithms address the ground target coverage problem using a perspective camera model. Other studies [8–11] have considered optimal placement of camera-mounted drones with the aim of minimizing total energy consumption while maximizing target coverage. Again, these efforts considered a perspective camera model, rather than an omnidirectional or heterogeneous stereo vision setup.

The studies above focused on ensuring ground target coverage, rather than minimizing localization error, and adopted a relatively simple camera model. Other studies on optimal placement have emphasized target observation using a variety of visual sensors. References [12–14] propose algorithms that find optimal positions for ground-based mobile cameras to maximize the aggregate observability of objects' motion in a confined area. These papers adopt a pinhole camera model. Other work has addressed the problem of placing multiple cameras for 3D reconstruction [15–17] and 3D motion capture [18]. These papers focus on minimizing depth estimation error, but the applications involve indoor operation where the objects being images are a short distance away. A number of papers have provided detailed analyses of triangulation-based localization error [19–21], illustrating how this error is affected by both the location of sensors and the line-of-sight (LOS) pointing angles from the sensors to a target. These results were then used to develop optimal sensor placement algorithms to minimize the target localization error using ground-based robots. Reference [22] discusses triangulation-based localization accuracy using two omnidirectional cameras to estimate the optimal positions.

A few studies have considered optimal placement of cameras with different fields of view. Reference [23] defines various types of camera models to solve an area coverage problem. Similarly, reference [24] addresses a target coverage problem using perspective cameras with different FOVs. Reference [25] describes the use of omnidirectional and perspective cameras to cover an area with the minimum number of cameras.

While a number of studies have considered optimal drone/camera placement to cover an area, or to minimize the depth estimation error, none have addressed the localization of threat aircraft using multiple aircraft equipped with heterogeneous camera systems.

III. Camera models and triangulation-based localization

We assume the host aircraft is equipped with the PCV system described in [3]. The system includes two cameras: a peripheral vision camera and a central vision camera. The peripheral vision camera comprises two 180°-FOV fisheye

lenses to provide a large FOV. The central vision camera has a 53° FOV and is capable of pan-tilt operation. Table 1 shows the specifications for the two cameras. Although we consider a specific camera setup in order to describe numerical and experimental results, the analysis described here can be easily extended to a large class of cameras.

Table 1 System parameters

Parameter	Peripheral camera	Central camera
Focal length (mm)	1.0	4.8
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

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

A. Peripheral vision camera model

For the peripheral vision camera model, we use a fisheye lens model available in the OpenCV library [26, 27] which accounts for the refraction of light rays passing through the lens, an effect called lens distortion. As an example, Figure 1 illustrates how objects at the edge of the image are highly distorted so that they appear smaller than if they appeared at the center of the image.

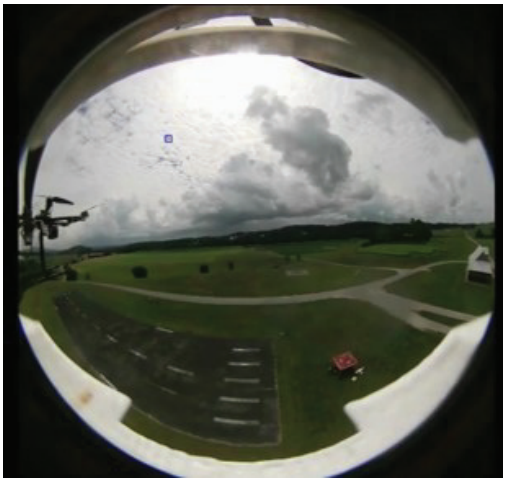


Fig. 1 Lens distortion of a fisheye lens

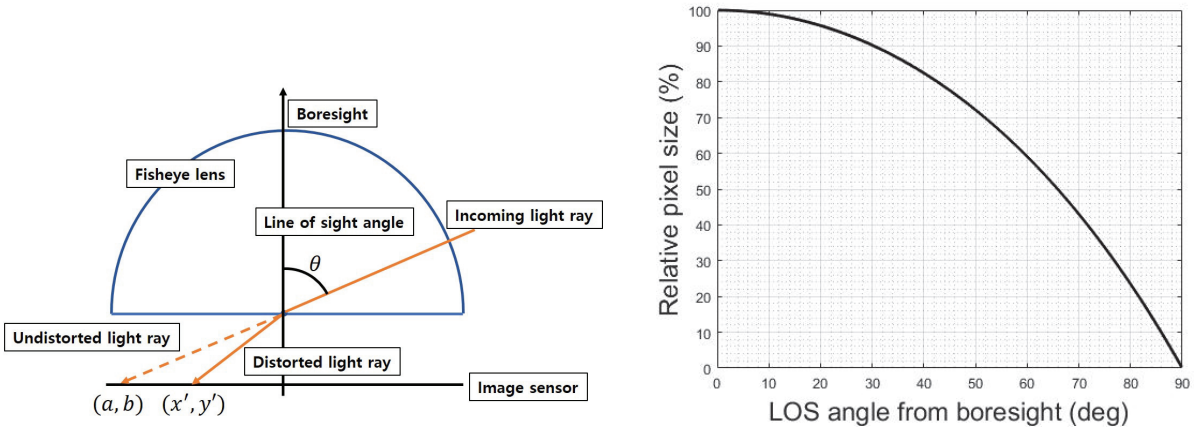
Computer vision based detection algorithms require that the detected object appear in the image with some minimum number of associated pixels. For example, optical flow requires at least 25 px (5 px wide × 5 px high) to detect a moving object in the image. However, the pixel size of the object on the image can differ depending on the line of sight (LOS) angle to the object. An object of interest that occupies 25 px at the center of the image would occupy fewer pixels at the edge of the image, and hence be undetectable using optical flow. The pixel size of a threat aircraft on the fisheye camera image should be larger than the minimum number of pixels that the detection algorithm requires for a detection. Following is a model for the distortion of a light ray that enters the fisheye lens with line of sight (LOS) angle θ relative to the camera boresight, as shown in Figure 2(a):

$$\theta_d = \theta(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8) \tag{1}$$

The parameters k_i for $i \in \{1, 2, 3, 4\}$ are the lens distortion coefficients obtained through a camera calibration process [26, 27]. The (distorted) light ray is then projected onto the image pixel coordinates u and v:

$$u = \frac{\theta_d}{\tan\theta}a \qquad \text{and} \qquad v = \frac{\theta_d}{\tan\theta}b \tag{2}$$

where a and b represent the pixel coordinates if the incoming light ray were not distorted by the lens. These equations illustrate how a larger LOS angle results in smaller pixel sizes for objects near the edge of the image. Figure 2(b) shows the relative pixel size of an object in the image versus the LOS angle. At the lens center, there is no distortion, so the relative pixel size is 100%; the relative pixel size decreases to 0% at a 90° LOS angle.



(a) Fisheye distortion geometry

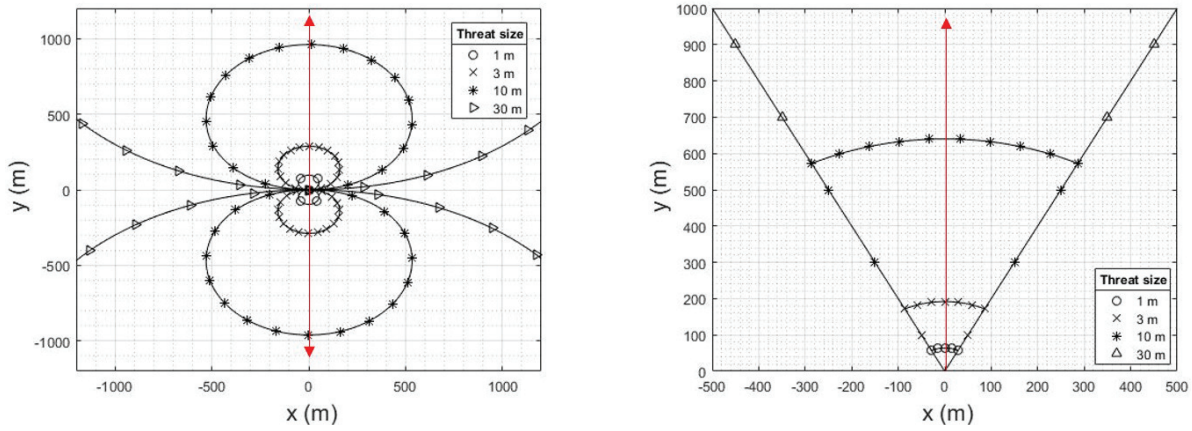
(b) Fisheye pixel density against FOV angle

Fig. 2 Fisheye lens distortion

The pixel size of an object increases when the object moves closer to the camera. We define the *detectable region* for a given camera with respect to an object of given size and location relative to the camera boresight. The pixel width of an object represented in the image can be estimated as:

$$\alpha_{\mathrm{p}} = \frac{f^{\mathrm{p}} W_{\mathrm{t}} \gamma^{\mathrm{p}} \theta_d}{d_{\mathrm{t}} W_s^{\mathrm{p}} \tan\theta} \tag{3}$$

where W_{t} is the width of the actual object (the "threat"), f^{p} is the focal length of the peripheral vision camera, γ^{p} is the horizontal resolution (in px) of the peripheral vision camera, d_{t} is the distance to the object, and W_s^{p} is the (physical) width of the peripheral vision camera's sensor. The same equation can be used for the pixel height of the object. The pixel width and height should be larger than the minimum number of pixels required by the detection algorithm.



(a) Detectable region of the peripheral vision camera (red arrows: boresight of two fisheye lenses)

(b) Detectable region of the central vision camera (red arrow: boresight of lens)

Fig. 3 Detectable region of two cameras

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

Figure 3(a) shows regions in which objects of different size can be detected by the peripheral vision camera using optical flow. (A qualitatively similar figure would result for any fisheye camera; the specific bounding curves depend on the given camera parameters.) As the peripheral vision camera has two fisheye cameras, there are two boresight directions, which are indicated as arrows in the figure. The plot illustrates the detectable region for an object that is at least as large as indicated in the legend. The figure illustrates the role of object size and LOS angle in the detectability of an object.

B. Central vision camera model

The central vision camera is a perspective camera. The lens distortion is small and the projection error can be corrected through camera calibration. The pixel size of an object at a given distance therefore remains constant throughout the camera's FOV, in contrast to the peripheral vision camera. We adopt a pinhole camera model for the central vision camera. As shown in Figure 4, the pinhole camera model assumes no lens distortion, so the pixel width of an object is proportional to its size:

$$\alpha_{\mathrm{c}} = \frac{f^{\mathrm{c}} W_{\mathrm{t}} \gamma^{\mathrm{c}}}{d_{\mathrm{t}} W_s^{\mathrm{c}}} \tag{4}$$

where f^{c} is the focal length of the central vision camera, γ^{c} is the horizontal resolution of the central vision camera, W_s^{c} is the width of the central vision camera sensor. The threat pixel height can be computed in the same way. The pixel width and height of an object must be larger than the minimum number of pixels needed by the central vision detection algorithm.

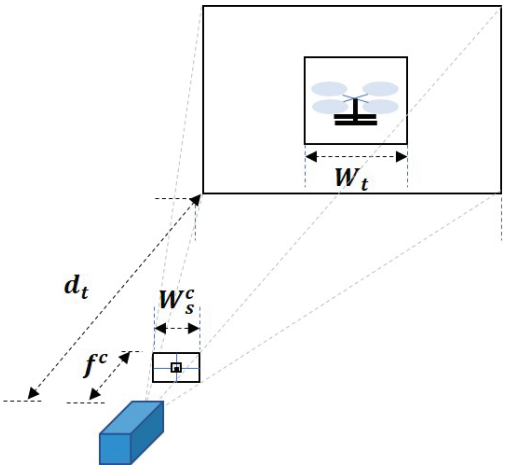


Fig. 4 The central vision camera model geometry

Because the central vision camera provides a high resolution image, the pixel size of an imaged object may be sufficient to enable classification using a deep neural network algorithm such as YOLO [28, 29]. We assume that YOLO is used for central vision detection as described in [3]. Earlier work indicates that YOLO requires at least 900 px (30 px wide × 30 px high) to detect a threat aircraft, so the threat pixel width and height should be larger than 30 px. The detectable region of the central vision camera when YOLO is used for detection is illustrated in Figure 3(b). Also, because the FOV of the central vision camera is limited, the bearing angle between the camera boresight and the object should be no more than half the central vision camera's FOV angle.

C. Triangulation-based localization and error

In previous work [3], we described a heterogeneous stereo vision algorithm for threat aircraft localization using the two cameras described in the preceding subsections. Once a threat aircraft is detected in a camera image, a three-dimensional vector is defined that points toward the threat aircraft in the camera-fixed reference frame. If two of these "threat vectors" are obtained simultaneously from two cameras at different, known locations, we may use them

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

to triangulate the threat's position in the global reference frame (denoted "g"):

$$\vec{r}_{\mathrm{t/g}} = \vec{r}_{\mathrm{p/g}} + \frac{\left\|\vec{r}_{\mathrm{t/c}} \times \vec{r}_{\mathrm{p/c}}\right\|}{\left\|\vec{r}_{\mathrm{t/c}} \times \vec{r}_{\mathrm{t/p}}\right\|} \vec{r}_{\mathrm{t/p}} \tag{5}$$

where $\vec{r}_{\mathrm{t/p}}$ and $\vec{r}_{\mathrm{t/c}}$ are the threat vectors in the peripheral vision camera-fixed reference frame (denoted "p") and in the central vision camera-fixed reference frame (denoted "c"), respectively, and where $\vec{r}_{\mathrm{p/g}}$ and $\vec{r}_{\mathrm{p/c}}$ represent the positions of the optical centers of the peripheral and central vision cameras, respectively, in the global frame. (All vectors in (5) are assumed to be expressed in the global frame.) Figure 5 shows the geometry of heterogeneous stereo vision-based threat localization.

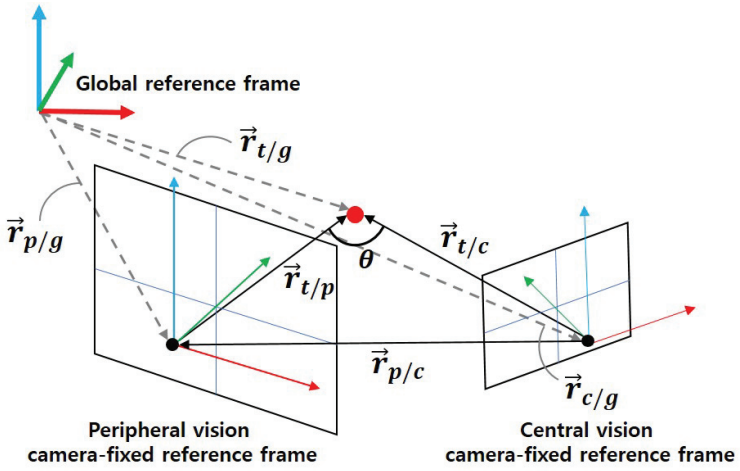


Fig. 5 Geometry of threat position estimation

The threat localization accuracy differs depending on the camera location and the threat distance. If the threat is close, the localization error will decrease; if the distance between the two cameras (the baseline) is too small, the localization error will increase. Reference [19] gives an expression for this triangulation-based localization error:

$$\delta_r = \frac{\left\|\vec{r}_{\mathrm{t/p}}\right\| \left\|\vec{r}_{\mathrm{t/c}}\right\|}{\sin\theta} \delta_s \tag{6}$$

where δ_s is the magnitude of the error in the threat location $\vec{r}_{\mathrm{t/g}}$ and θ is the intersection angle between the threat vectors from the two cameras:

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

See Figure 5. Equation (6) implies that localization error is minimum when the distance from the cameras to the threat is small and when the threat vectors intersect at a right angle.

IV. Optimal placement algorithm for multiple heterogeneous stereo vision systems

Considering the two camera models discussed in the previous section, this section describes a heterogeneous stereo-vision optimal placement (HSOP) algorithm. The algorithm determines the optimal positions for two or more host aircraft, each equipped with a PCV system, as well as the type of camera to be used, in order to cover the current position (given) and final position (predicted over a given time horizon) of one or more threat aircraft, minimizing the cumulative localization error. The host aircraft are assumed to take the computed positions and to hover there, imaging the threat aircraft over the given time horizon, after which the optimal host positions are updated by re-applying the HSOP algorithm. The host aircraft continue tracking the threat aircraft until they are no longer a threat. Note that the focus of this note is the placement of sensors for optimal localization – the "fix" portion of the "find, fix, finish" chain in a counter-UAS strategy.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

The output of the HSOP algorithm can thus be considered as a waypoint generation scheme for host aircraft. Motion of the host aircraft can degrade the performance of camera-based threat detection algorithms. While there are ways to compensate for ownship motion, such as physical or software image stabilization [3], observing and tracking a threat from a static position produces higher quality imagery which improves detection and localization performance.

This section presents the HSOP algorithm in detail. The algorithm requires the initial state (position and velocity) for each threat aircraft. While this requirement may seem circular, the initial threat state might be obtained (with lower accuracy) using a single PCV-equipped host that serves as a counter-UAS sentry in a given region, as described in [3]. The optimization problem is formulated under additional simplifying assumptions outlined below, though each of these can be relaxed with some effort:

- All host and threat aircraft fly at the same altitude.
- The number of available host aircraft is sufficient to cover all threat aircraft.
- The threat aircraft follow a straight path at constant speed (i.e., they are non-cooperative, but non-antagonistic).
- The path traveled by a threat aircraft during the observation time horizon is contained within the detectable range of the imaging sensors.
- Host aircraft can instantly attain waypoints indicated by the HSOP algorithm (i.e., the physical placement of host aircraft occurs much faster than the optical motion of threat aircraft).
- The host aircraft can communicate with sufficient speed and volume to cooperatively localize threats.
- All PCV systems are identical; see Table 1.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

A. Initial feasibility check

Given initial states for each threat aircraft in a given set T, two host aircraft h_i are assigned to each threat aircraft $t_j \in T$ to enable triangulation. In this way, a *host-threat pair* s comprising the threat and the two assigned host aircraft is generated for each threat aircraft. As an example, suppose there are two threat aircraft and three host aircraft. We first assign the nearest unassigned host to each given threat as an "initial observer" and label that host accordingly: (h_1, t_1) and (h_2, t_2). We then complete the pairs by assigning a second host aircraft to each threat. A *host-threat pair set* (or just *pair set*) S comprises one host-threat pair for each threat aircraft. For the given example, one possible pair set is $S = \{(t_1, h_1, h_2), (t_2, h_2, h_1)\}$. In this case, only the two host aircraft (h_1, h_2) are used to observe both of the two threat aircraft. Given three available host aircraft for the two detected threats, there are three more possible pair sets: $\{(t_1, h_1, h_2), (t_2, h_2, h_3)\}, \{(t_1, h_1, h_3), (t_2, h_2, h_1)\}, \{(t_1, h_1, h_3), (t_2, h_2, h_3)\}$.

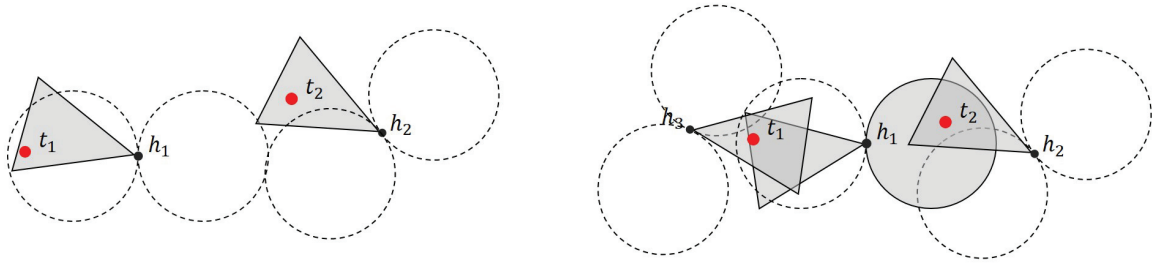
If the number of host and threat aircraft is large, the number of possible pair sets scales as $(N_\mathrm{h} - 1)^{N_\mathrm{t}}$, where N_h is the number of the host aircraft and N_t is the number of threat aircraft. To ameliorate this scaling issue, we use a backtracking approach to exclude infeasible pair sets. Algorithm 1 below presents pseudo-code for generating pair sets. Algorithm 2 then culls this set by excluding infeasible pair sets.

Algorithm 1: `Generate-Pair-Sets`(t_i, N_h, SP)

```
tmpSet = {};
for i to N_h do
    // h_f : initial observer of t_i
    if h_i ≠ h_f then
        tmpSet.insert({t_i, h_f, h_i})
SP ← Generate-Combinations(tmpSet,SP);
return SP;
```

In Algorithm 1, SP indicates the set of pair sets, `Generate-Combinations` is a combination generation function that can be implemented using, for example, the `nchoosek` function in MATLAB. This function makes combinations of sets in SP and pairs in `tmpSet` and adds the resulting sets to SP.

Next, the feasibility of pair sets in SP is determined. An infeasible pair set is one that includes host-threat pairs which are physically impossible. Consider Figure 6(a), for example, where circles and triangles illustrate the detectable regions of peripheral and central vision cameras,respectively; recall Figures 3(a) and 3(b). (Dashed lines indicate cameras that are not used for localization.) For the given configuration, the pair set $\{(t_1, h_1, h_2), (t_2, h_2, h_1)\}$ is infeasible because images from at least two host aircraft are needed to triangulate each threat and t_1 and t_2 cannot be covered by h_1 and h_2 simultaneously. On the other hand, the pair set $\{(t_1, h_1, h_3), (t_2, h_2, h_1)\}$ depicted in Figure 6(b)



(a) Infeasible host-threat pair set

(b) Feasible host-threat pair set

Fig. 6 Examples of infeasible and feasible host-threat pair sets

is feasible.

A simple initial screen for feasibility involves checking whether the assigned threats can be placed within the detectable region of a candidate host's peripheral or central vision camera. In the example of Figure 6(a), two threat aircraft are assigned to h_1 and h_2, but the distance between two threats is large so that the detectable region of a peripheral vision camera cannot cover both threats at once. In this case, at least four hosts are required to cover each threat from two perspectives. On the other hand, the two threats in Figure 6(b) are closer, which enables h_1 to cover both threats simultaneously; only three hosts are required to cover each threat from two perspectives. In the example of Figure 6, the threat aircraft are static. For a moving threat t_i, the assigned host aircraft must be able to cover the path connecting the current threat position $\vec{x}_{t_i}$ to the predicted position after some specified time τ:

$$\vec{x}_{t_i}^{\tau} = \vec{x}_{t_i} + \tau \vec{v}_{t_i} \tag{7}$$

where $\vec{v}_{t_i}$ is the threat velocity vector. Rather than require continuous coverage, however, we adopt a simpler strategy and add the final threat position to the pair s as if it is another threat aircraft to be included in the optimization process. (Given an initial threat pairing $s_1 = (t_1, h_1, h_2)$, for example, we define an *augmented* threat pairing $s_1^{\tau} = (t_1, t_1^{\tau}, h_1, h_2)$, where t_1^{τ} denotes the location of threat t_1 after time τ has elapsed.)

For each host h_i, a set T^{h_i} is defined which contains all assigned threats, at both their initial and predicted final positions. The algorithm checks if all threat positions in T^{h_i} can be placed inside the detectable range of the peripheral vision camera (e.g. using the `inpolygon` function in MATLAB). If this is possible for all h_i in the pair set, the pair set is declared feasible. For a feasible pair set, the camera type of each host in the pair set is next determined. If the threats assigned to h_i can all be contained within the detectable region of the central vision camera, then this camera type is chosen for h_i. If they cannot, then the peripheral vision camera is chosen. Even if a host aircraft's peripheral vision camera can cover all assigned threats, it may also be possible to cover one of these threats using the central vision camera, as shown in Figure 6. In this case, because of its greater resolution, the central vision camera is used to observe the threat. If multiple threats can be covered by the central vision camera, then the closest threat is observed by the central vision camera. Pseudo-code for this process is given in Algorithm 2. This initial feasibility check speeds the backtracking process described in the following section.

Algorithm 2: `Feasibility-Check`(N_h, T, SP)

```
for t_i in T do
    SP ← Generate-Pair-Sets(t_i, N_h, SP);
    for each pair set S from SP do
        if S is feasible then
            Determine-Camera-Type(S)
        else
            SP.erase(S)
return SP;
```

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

B. Backtracking process to remove infeasible branches

As mentioned in the previous section, the possible number of pair sets is $(N_\mathrm{h}-1)^{N_\mathrm{t}}$. It is expedient to remove all remaining host-threat pair sets that are infeasible before formulating and solving the optimal placement problem. A backtracking approach is used to complete the culling of infeasible pair sets. Backtracking is a constructive algorithm for generating feasible solutions. The process can be visualized as a tree structure in which candidate solutions are the branches that remain after pruning infeasible branches that do not satisfy constraints. Figure 7 shows an example of the backtracking process when there are three threat aircraft (t_1, t_2, t_3) and four host aircraft (h_1, h_2, h_3, h_4). Each column shows possible host-threat pairs for each threat aircraft, starting with threat t_1 at the left. In this case, there are 27 possible combinations for pair sets. Infeasible pair sets are removed through the backtracking process. First, the feasibility check is conducted only for pair combinations of t_1 and t_2. There are 9 possible combinations involving t_1 and t_2 pairs. If one of those combinations is infeasible, the combination is excluded (pruned). In Figure 7, for example, the combination $\{(t_1, h_1, h_2), (t_2, h_2, h_1)\}$ is declared infeasible because of the threats t_1 and t_2 are too distant, as in Figure 6(a). Downstream combinations involving threat t_3 are generated only for feasible host-threat pairings.

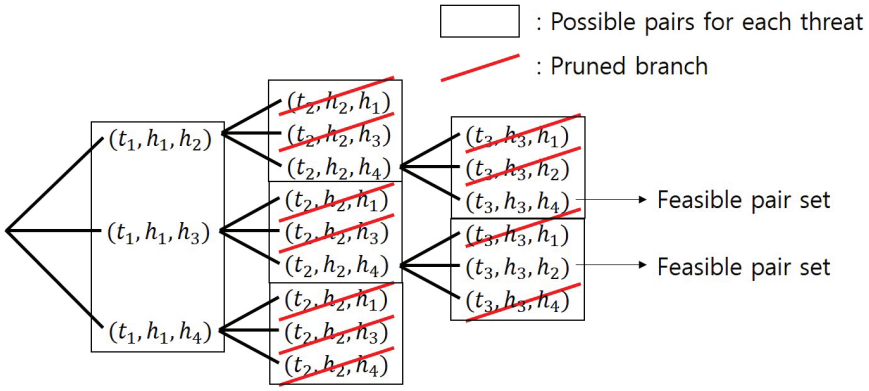


Fig. 7 Example of backtracking process

C. Optimization problem statement

Once feasible pair sets are chosen from the backtracking process described in the previous section, the optimization problem is solved for each pair set S. Recall that a host-threat pair s consists of a threat aircraft and two assigned host aircraft and that a pair set S contains a complete set of host-threat pairs, i.e., one and only one pair for each threat. The aim is to minimize the localization error (the sum squared error between the ground truth threat position vector and the estimated threat position vector) of all threat aircraft in a pair set S. The camera type of the host aircraft h_i is distinguished in the formulation using a binary value $c_\mathrm{p}^{h_i}$. If the camera type to be used by host h_i is the peripheral vision camera, then $c_\mathrm{p}^{h_i} = 1$; if the camera type to be used is the central vision camera, $c_\mathrm{p}^{h_i} = 0$. As described in Section III.C, the intersection angle $\theta_s^{t_j}$ of two threat vectors should be close to 90° to minimize the triangulation-based localization error and the distance $d_{t_j}^{h_i}$ to the threat from the host aircraft should be small. Also, the pixel width ($\alpha_\mathrm{p}^{h_i t_j}$ and $\alpha_\mathrm{c}^{h_i t_j}$) of the threat aircraft in the peripheral and central vision imagery of host h_i should be larger than the minimum number of pixels (γ_p and γ_c) required by the peripheral and central vision detection algorithms to ensure detection. To simplify analysis, we assume the width and height of all threat aircraft are equal and that all aircraft operate at the same altitude. In this case, we may consider only the threat pixel width in the formulation. Finally, the optimization problem to find the set X_h of host position vectors which minimizes the localization error of all threat aircraft is formulated below.

$$X_h^* = \arg\min_{X_h} \quad \sum_{s\in S}\sum_{t_j\in s}\left\{\frac{\prod_{h_i\in s}\left\{\left[c_{\mathrm{p}}^{h_i}/\alpha_{\mathrm{p}}^{h_i t_j} + (1-c_{\mathrm{p}}^{h_i})/\alpha_{\mathrm{c}}^{h_i t_j}\right] d_{t_j}^{h_i}\right\}}{\sin\theta_s^{t_j}}\right\} \tag{8}$$

$$\text{s.t.} \quad (c_{\mathrm{p}}^{h_i}\alpha_{\mathrm{p}}^{h_i t_j} - \gamma_{\mathrm{p}}) + (1-c_{\mathrm{p}}^{h_i})(\alpha_{\mathrm{c}}^{h_i t_j} - \gamma_{\mathrm{c}}) > 0, \qquad h_i \in s, \;\; t_j \in s, \;\; s \in S \tag{9}$$

$$(1-c_{\mathrm{p}}^{h_i})\left(\frac{\psi_{\mathrm{c}}^{h_i}}{2} - |\psi_{t_j}^{h_i}|\right) \geq 0, \qquad h_i \in s, \;\; t_j \in s, \;\; s \in S \tag{10}$$

$$d_{t_j}^{h_i} \geq R_{\mathrm{s}}, \qquad h_i \in s, \;\; t_j \in s, \;\; s \in S \tag{11}$$

$$c_{\mathrm{p}}^{h_i} \in \{0,1\}, \qquad h_i \in s, \;\; s \in S \tag{12}$$

where $\psi_{\mathrm{c}}^{h_i}$ is the FOV angle of the central vision camera for host h_i, $\psi_{t_i}^{h_i}$ is the azimuth angle between h_i and t_i, and R_{s} is a safety distance between host and threat aircraft to prevent collisions. (We assume that collisions between hosts and maneuvering threats are prevented by an over-riding collision avoidance protocol. The objective function (8) determines the host locations that maximize the pixels in the image that correspond to the threat aircraft and minimize the error in (6). Constraint (9) ensures the minimum number of "threat pixels" for detection. Constraint (10) ensures that the threat is within the FOV of the central vision camera, if that camera is to be used. The minimum distance between the host aircraft and the threat aircraft is constrained to be larger than the safety distance R_{s} in (11). The optimization problem defined above is solved using the interior-point method for each feasible pair set. An example pair set is illustrated in Figure 8.

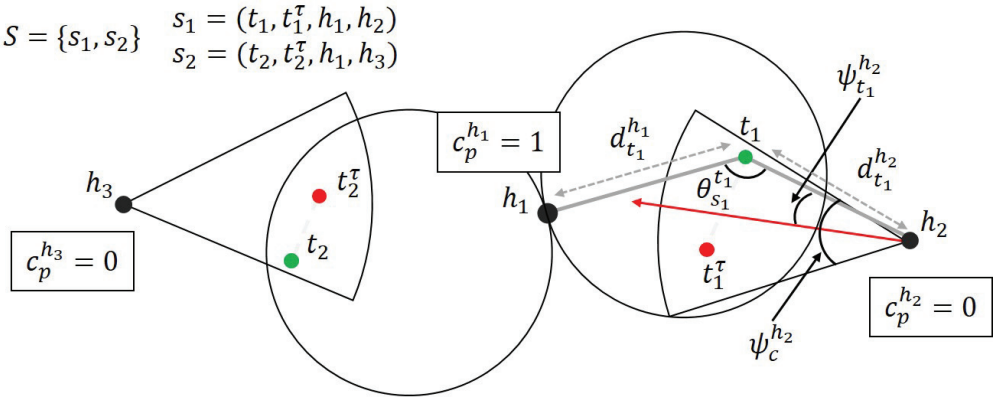


Fig. 8 Geometry of a pair set

The HSOP algorithm is given in Algorithm 3. To minimize the number of host aircraft involved, the HSOP algorithm initially generates pair sets using only initial observers ($N_{\mathrm{h}} \leq N_{\mathrm{t}}$). If there exists no feasible pair set in SP, the algorithm adds host aircraft until a feasible pair set is found. (Note that the maximum number of hosts N_{h} required to image N_{t} threats is $2N_{\mathrm{t}}$.) Once a feasible set is found, the optimization problem is solved using the pair set. If the cost J of the resulting placement set X_h is smaller than the minimum cost seen so far, and if X_h is determined to satisfy all constraints (9)-(12) using the `Constraints-Check` function, then the optimal placement set X_h^* and cost J^* are updated accordingly. The resulting X_h^* indicates the optimal locations for the minimum number of host aircraft which cover the current and final positions of all threat aircraft and which also minimize the cumulative threat localization error, including current and final threat positions. In the next iteration, final threat state estimates are taken as initial states and the HSOP algorithm computes the new optimal positions for the host aircraft.

V. Results

Because of the logistical challenges associated with multi-host, multi-threat detection and localization experiments, simulations were used to assess the performance of the HSOP algorithm. In these simulations, the HSOP algorithm is used to compute host aircraft positions corresponding to randomly generated threat states (positions and velocities). The localization error for each threat aircraft position is then estimated. (The localization error is the sum squared

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

Algorithm 3: `HSOP algorithm`(N_h, T)

$SP = \{\}$;
while X_h^* is empty **do**
  $SP \leftarrow$ `Feasibility-Check`(N_h, T, SP);
  if SP is not empty **then**
    for each pair set S in SP **do**
      $[X_h, J] \leftarrow$ `Solve-Problem`(S);
      if `Constraints-Check`(X_h) & $J < J^*$ **then**
        $X_h^* \leftarrow X_h$;
        $J^* \leftarrow J$;
  else
    $N_h \leftarrow N_h + 1$
return X_h^*;

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

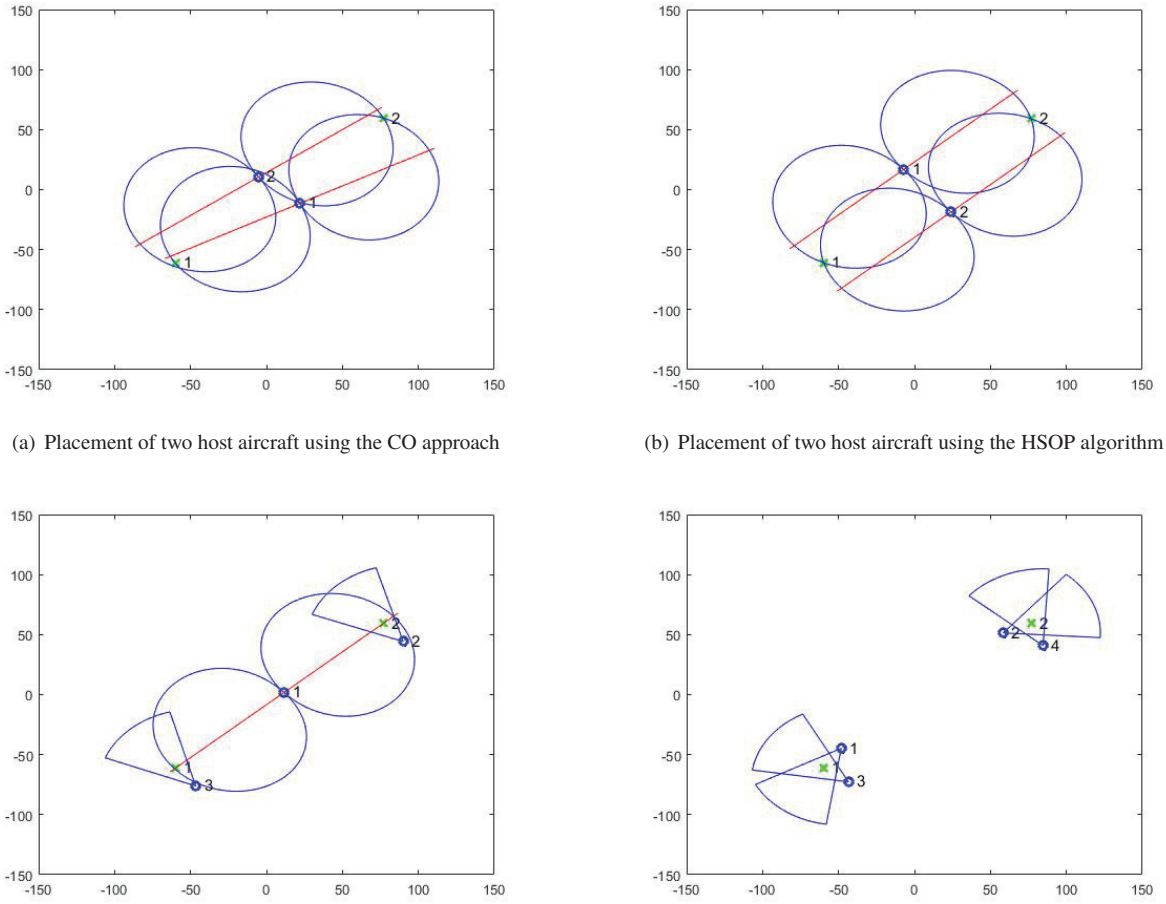
error between the ground truth threat position and the estimated threat position.) The simulation parameters are shown in Table 2. To simulate the localization error, we superimpose zero-mean Gaussian noise to the threat image in the horizontal and vertical directions with a 30 px standard deviation for the peripheral vision camera and a 3 px standard deviation for the central vision camera. (This empirically tuned synthetic error closely matches the error seen in experimental measurements.) The localization error is then estimated using the noise-corrupted threat vectors. This process is repeated 10,000 times for each simulation and the localization error is averaged. We concluded that 10,000 data points are enough to assess the localization error with the assumed standard deviation because a similar averaged error is obtained using 1000 data points. References [4–11] provide host aircraft placement algorithms for threat coverage, but localization error of threats is not considered in determining host placement. To illustrate how well the HSOP algorithm reduces localization error compared with algorithms that consider only threat coverage, we compare the HSOP algorithm with such a "coverage only (CO)" approach obtained by eliminating the localization cost from the HSOP algorithm and using the same constraints (9)-(12).

Table 2 Simulation parameters

Parameter	Value
Threat speed (m/s)	0 - 5
Prediction horizon (sec)	10
R_s (m)	20
γ_p (px)	5
γ_c (px)	30
$\psi_c^{h_i}$ (°)	53

Figure 9 shows results of the CO approach and the HSOP algorithm in a sample simulation using two static threat aircraft. In the CO approach, shown in Figure 9(a), the algorithm successfully covers the two threat aircraft using two host aircraft, but this algorithm does not consider the localization error that results when the hosts triangulate the position of the threats. The HSOP algorithm, whose results are shown in Figure 9(b), simultaneously covers the threat aircraft, with a minimum distance to the threats, and maximizes the intersection angles of the threat vectors from the host aircraft to the two threats.

As mentioned in the previous section, the HSOP algorithm provides positions for the minimum number of host aircraft. In this example scenario, the minimum number of host aircraft needed is 2. To see how the localization error changes if more host aircraft are allowed, we ran the HSOP algorithm using $N_h = 3$ and 4 and again estimated the localization error. Figures 9(c) and 9(d) show the results of the HSOP algorithm when there are three and four host aircraft, respectively. Note that type of camera selected from the hosts' PCV systems changes to minimize the



(a) Placement of two host aircraft using the CO approach

(b) Placement of two host aircraft using the HSOP algorithm

(c) Placement of three host aircraft using the HSOP algorithm

(d) Placement of four host aircraft using the HSOP algorithm

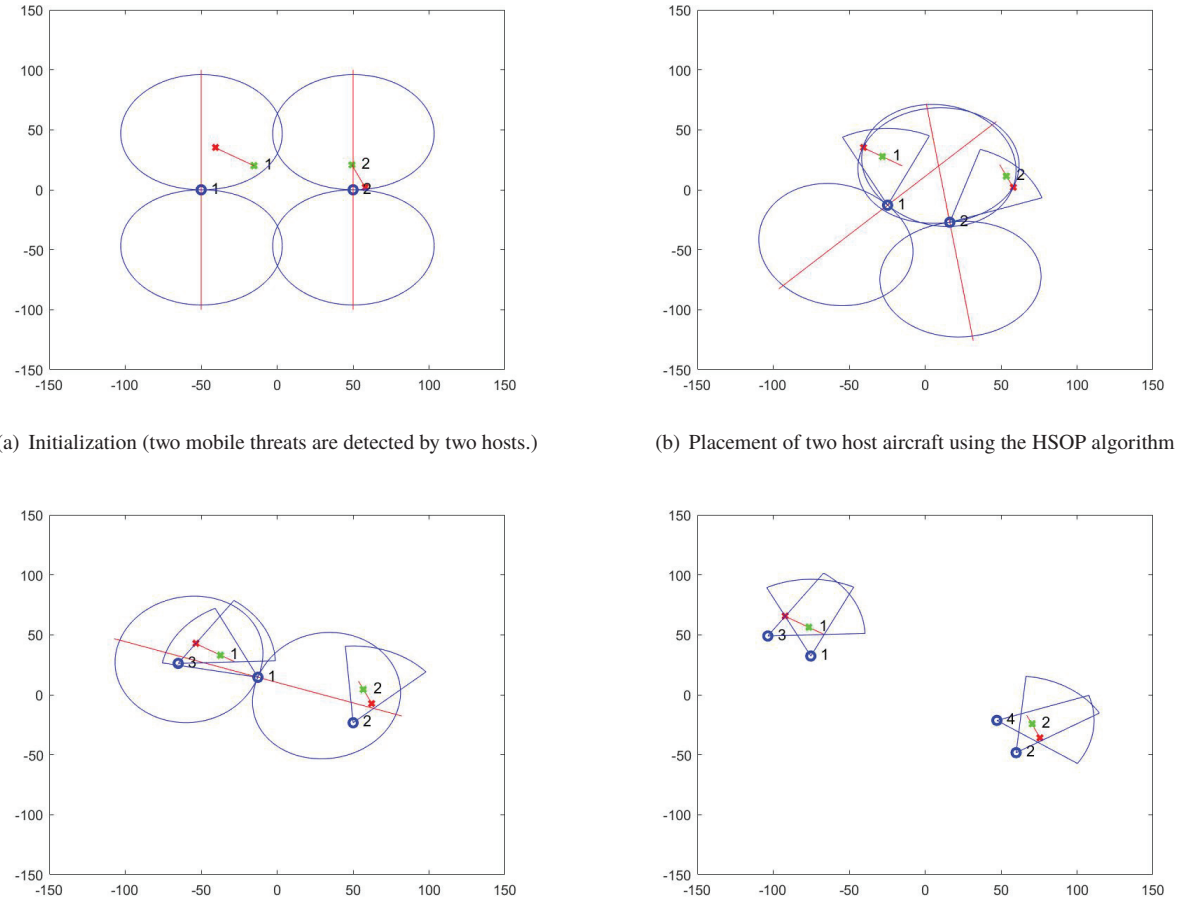
Fig. 9 Simulation results for optimal host aircraft placement using two approaches for static threats. (Small blue circles: host aircraft positions. Green x marks: threat positions. Red lines: boresight directions of peripheral vision cameras. Large blue circles: detectable regions of peripheral vision cameras. Blue circular sectors: detectable regions of central vision cameras.)

localization error. Also note that the intersection angles are 90°.

Figure 10 shows results of the HSOP algorithm with two mobile threat aircraft. The threats are initially detected by two host aircraft and their future path is predicted based on the initial detection data. The HSOP algorithm successfully covers the entire path of the threat aircraft throughout the prediction horizon and optimizes the host position to minimize the localization error. Once the prediction time (τ) has elapsed, a new final position is computed corresponding to the updated threat position and velocity. The HSOP algorithm then generates new host positions based on the new threat paths. Note that the camera types and the number of hosts needed change as the threats move.

To better appreciate the localization error obtained using the HSOP algorithm, the simulation is repeated 100 times with different threat positions and various numbers of host aircraft. The localization error of all threat aircraft is estimated in each simulation and averaged. Figure 11 shows the averaged localization error using the CO approach and using the HSOP algorithm versus the number of host aircraft. The three subfigures show that the localization error of the HSOP algorithm decreases with increasing numbers of host aircraft. When fewer host aircraft are available, the peripheral vision cameras must be used to cover all of the threat aircraft, as shown in Figures 9 and 10. If more host aircraft are available, the central vision camera is selected by the algorithm and the localization error decreases because of the camera's higher resolution. On the other hand, the localization error using the CO approach does not change with the number of host aircraft since the CO approach only considers the threat coverage.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393



(a) Initialization (two mobile threats are detected by two hosts.)

(b) Placement of two host aircraft using the HSOP algorithm

(c) Placement of three host aircraft using the HSOP algorithm

(d) Placement of four host aircraft using the HSOP algorithm

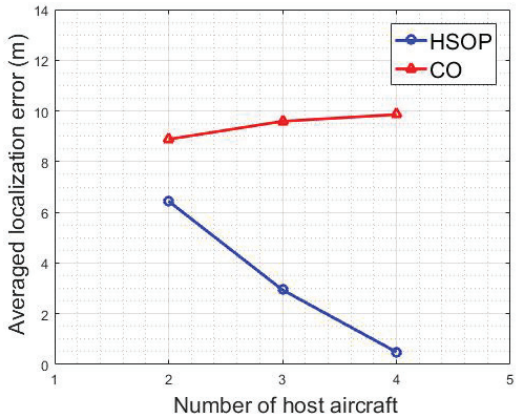
Fig. 10 Simulation results for optimal host aircraft placement using the HSOP algorithm for mobile threats. (Small blue circles: host aircraft positions. Green x marks: current threat positions. Red x marks: final threat positions. Red lines: boresight directions of peripheral vision cameras. Large blue circles: detectable regions of peripheral vision cameras. Blue circular sectors: detectable regions of central vision cameras.)

VI. Conclusions

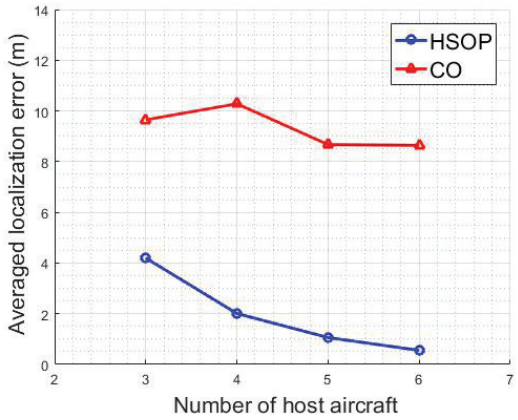
This note suggests an algorithm for heterogeneous stereo vision system placement for airborne counter-UAS applications, a variant on the well-studied problem of placing drones for optimal coverage of ground targets. The algorithm ensures that host aircraft equipped with two different types of camera – an omnidirectional "peripheral vision" camera and a perspective view "central vision" camera – can cover and localize a set of threat aircraft with minimum localization error. Simulation results illustrate that the heterogeneous stereo-vision optimal placement (HSOP) algorithm reduces localization error. For the specific aircraft and sensing systems considered here, the HSOP algorithm reduced the averaged localization error to less than 1 m in simulations, compared with a 9-10 m localization error associated with a coverage-only approach.

In considering the optimal placement of drones to localize moving threats, we have assumed the host aircraft are sufficiently fast to attain the commanded positions quickly relative to threat aircraft motion. This assumption may be unrealistic in some scenarios. Future work might address this assumption in terms of optimal motion planning, although the image processing quality may suffer from camera motion. Also, while simulations suggest that the HSOP algorithm is effective, in practice, the algorithm does not always provide a globally optimum solution nor is it guaranteed to converge. One might define a heuristic method to initialize the optimization problem in order to avoid these issues. Another potential concern is the algorithm's computational complexity. Although we prune infeasible

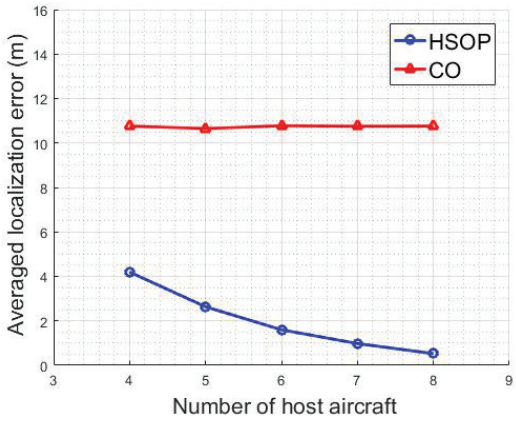
Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393



(a) Localization error with two threat aircraft



(b) Localization error with three threat aircraft



(c) Localization error with four threat aircraft

Fig. 11 Localization error using CO and HSOP algorithm

host configurations to speed up the algorithm, if the number of host and threat aircraft is large (of order 10, for example), the time required to compute a solution may prohibit its use for real-time operation. Ongoing work is aimed at relaxing some of the assumptions underlying the HSOP algorithm and making it run more efficiently.

Acknowledgments

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] Mueller, B., and Tsang, A., "At a busy airport in Britain, only pesky drones are flying," *The New York Times*, Dec. 20, 2018, p. A1.

[2] Michel, A. H., "Counter-drone systems 2nd edition," Center for the Study of the Drone at Bard College, 2019.

[3] Kang, C., Chaudhry, H., Woolsey, C. A., and Kochersberger, K. B., "Development of a peripheral-central vision system for small UAS tracking," *AIAA SciTech*, San Diego, California, January 7-11, 2019. https://doi.org/10.2514/6.2019-2074.

[4] Savkin, A. V., and Huang, H., "Proactive deployment of aerial drones for coverage over very uneven terrains: A version of the 3d art gallery problem," *Sensors*, Vol. 19(6), 2019, p. 1438. https://doi.org/10.3390/s19061438.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

[5] Tuba, E., Capor-Hrosik, R., Alihodzic, A., and Tuba, M., "Drone placement for optimal coverage by brain storm optimization algorithm," *In International Conference on Health Information Science*, Moscow, Russia, October 7-9, 2017, pp. 167–176. https://doi.org/10.1007/978-3-319-76351-4_17.

[6] Strumberger, I., Sarac, M., Markovic, D., and Bacanin, N., "Moth search algorithm for drone placement problem," *International Journal of Computers*, Vol. 3, 2018.

[7] Tuba, E., Tuba, I., Dolicanin-Djekic, D., Alihodzic, A., and Tuba, M., "Efficient drone placement for wireless sensor networks coverage by bare bones fireworks algorithm," *In 2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, Antalya, Turkey, March 22-25, 2018, pp. 1–5. https://doi.org/10.1109/ISDFS.2018.8355349.

[8] Zorbas, D., Razafindralambo, T., Pugliese, L. D. P., and Guerriero, F., "Energy efficient mobile target tracking using flying drones," *Procedia Computer Science*, Vol. 19, 2013, pp. 80–87. https://doi.org/10.1016/j.procs.2013.06.016.

[9] Pugliese, L. D. P., Guerriero, F., Zorbas, D., and Razafindralambo, T., "Modelling the mobile target covering problem using flying drones," *Optimization Letters*, Vol. 10(5), 2016, pp. 1021–1052. https://doi.org/10.1007/s11590-015-0932-1.

[10] Zorbas, D., Pugliese, L. D. P., Razafindralambo, T., and Guerriero, F., "Optimal drone placement and cost-efficient target coverage," *Journal of Network and Computer Applications*, Vol. 75, 2016, pp. 16–31. https://doi.org/10.1016/j.jnca.2016.08.009.

[11] Al-Turjman, F., Zahmatkesh, H., Al-Oqily, I., and Daboul, R., "Optimized unmanned aerial vehicles deployment for static and mobile targets monitoring," *Computer Communications*, Vol. 149, 2020, pp. 27–35. https://doi.org/10.1016/j.comcom.2019.10.001.

[12] Fiore, L., Somasundaram, G., Drenner, A., and Papanikolopoulos, N., "Optimal camera placement with adaptation to dynamic scenes," *In 2008 IEEE International Conference on Robotics and Automation*, Pasadena, California, May 19-23, 2008, pp. 956–961. https://doi.org/10.1109/ROBOT.2008.4543328.

[13] Bodor, R., Drenner, A., Janssen, M., Schrater, P., and Papanikolopoulos, N., "Mobile camera positioning to optimize the observability of human activity recognition tasks," *In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, August 2-6, 2005, pp. 1564–1569. https://doi.org/10.1109/IROS.2005.1545599.

[14] Bodor, R., Drenner, A., Schrater, P., and Papanikolopoulos, N., "Optimal camera placement for automated surveillance tasks," *Journal of Intelligent and Robotic Systems*, Vol. 50(3), 2007, pp. 257–295. https://doi.org/10.1007/s10846-007-9164-7.

[15] Chen, J., Khatibi, S., and Kulesza, W., "Planning of a multi stereo visual sensor system-depth accuracy and variable baseline approach," *In 2007 3DTV Conference*, Kos Island, Greece, May 7-9, 2007, pp. 1–4. https://doi.org/10.1109/3DTV.2007.4379391.

[16] Kulesza, W., Chen, J., Khatibi, S., and Bhatti, A., "Arrangement of a multi stereo visual sensor system for a human activities space," *Stereo Vision*, IntechOpen, 2008, pp. 153–172. https://doi.org/10.5772/5894.

[17] Malik, R., and Bajcsy, P., "Automated placement of multiple stereo cameras," *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, Marseille, France, October 17, 2008.

[18] Rahimian, P., and Kearney, J. K., "Optimal camera placement for motion capture systems in the presence of dynamic occlusion," *In Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, Beijing, China, November 13-15, 2015, pp. 129–138. https://doi.org/10.1145/2821592.2821596.

[19] Kelly, A., "Precision dilution in triangulation based mobile robot position estimation," *Intelligent Autonomous Systems*, 2003, pp. 1046–1053.

[20] Sanders-Reed, J. N., "Triangulation position error analysis for closely spaced imagers," *SAE Transactions*, Vol. 111, 2002, pp. 978–984. https://doi.org/10.4271/2002-01-0685.

[21] Sanders-Reed, J. N., "Impact of tracking system knowledge on multisensor 3D triangulation," *AeroSense 2002*, Orlando, Florida, July 1, 2002, pp. 33–41. https://doi.org/10.1117/12.472599.

[22] Shih, S. E., and Tsai, W. H., "Optimal design and placement of omni-cameras in binocular vision systems for accurate 3-D data measurement," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 23(11), 2013, pp. 1911–1926. https://doi.org/10.1109/TCSVT.2013.2269021.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393

[23] Altahir, A. A., Asirvadam, V. S., Hamid, N. H., Sebastian, P., Saad, N., Ibrahim, R., and Dass, S. C., "Modeling multicamera coverage for placement optimization," *IEEE Sensors Letters*, Vol. 1(6), 2017, pp. 1–4. https://doi.org/10.1109/LSENS.2017.2758371.

[24] Yildiz, E., Akkaya, K., Sisikoglu, E., and Sir, M. Y., "Optimal camera placement for providing angular coverage in wireless video sensor networks," *IEEE Transactions on Computers*, Vol. 63(7), 2013, pp. 1812–1825. https://doi.org/10.1109/TC.2013.45.

[25] Gonzalez-Barbosa, J. J., García-Ramírez, T., Salas, J., and Hurtado-Ramos, J. B., "Optimal camera placement for total coverage." *In 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17, 2009, pp. 844–848. https://doi.org/10.1109/ROBOT.2009.5152761.

[26] Bradski., G., "The OpenCV Library," *Dr. Dobbs Journal of Software Tools*, 2000.

[27] Bradski, G., and Kaehler, A. (eds.), *Learning OpenCV: Computer vision with the OpenCV library*, chapter and pages.

[28] Redmon, J., and Farhadi, A., "YOLO9000: better, faster, stronger," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, 2017, pp. 7263–7271. https://doi.org/10.1109/cvpr.2017.690.

[29] Redmon, J., and Farhadi, A., "YOLOv3: An Incremental Improvement," *arXiv preprint*, 2018.

Downloaded by VIRGINIA TECH on March 1, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2393