Multiple Cooperative Searchers for Aerial Surveillance

Satoshi Hoshino and Koya Kikuchi

Utsunomiya University, 7-1-2 Yoto, Utsunomiya, Tochigi 321-8585, JAPAN
hosino@cc.utsunomiya-u.ac.jp

Abstract. In this paper, we focus on an aerial surveillance system composed of single sentinel and multiple searchers. In the system, unidentified flight vehicles, UFVs, are surveillance targets. The searchers are required to approach the UFVs for identification. In this regard, however, faster UFVs move while avoiding the searchers. In order for the searchers to approach and detect such sophisticated UFVs, cooperative actions play an important role. Therefore, we apply reinforcement learning to the searchers. In the initial learning phase, however, searchers based on random actions seldom detect UFVs. Thus the searchers cannot acquire effective actions to approach the target UFV. Therefore, we further apply transfer learning to this problem. Through surveillance simulations, we show the effectiveness of the cooperative actions by the multiple searchers for the UFVs in the aerial surveillance system.

Keywords: multi-agent systems, reinforcement learning, cooperative actions, aerial surveillance

1 Introduction

In recent decades, unmanned aerial vehicles, UAVs, have been employed in several application domains, such as wildfire tracking, monitoring, search and rescue, and surveillance and reconnaissance [1][2]. In this paper, we focus on an aerial surveillance system composed of multiple UAVs. Basilico and Carpin have presented an aerial surveillance system composed of two types of UAVs [3], as illustrated in **Fig. 1**.

As the aerial agents, single sentinel and multiple searchers operate in the system. The sentinel operates at a place higher than the searchers. A sensing range of the sentinel is drawn by a pink circle. If an unidentified event occurs in the surveillance area, the sentinel detects the position. However, the sentinel is not capable of identifying the event precisely. On the other hand, searchers are able to identify the event in the sensing range drawn by red circles. Basilico and Carpin have succeeded in identifying events by dispatching the searchers.

For aerial surveillance with multiple UAVs, the area is divided and assigned to each of the UAVs [4]. In order to cover the assigned area, the UAV followed a path based on a deterministic zigzag pattern. On the other hand, cooperative

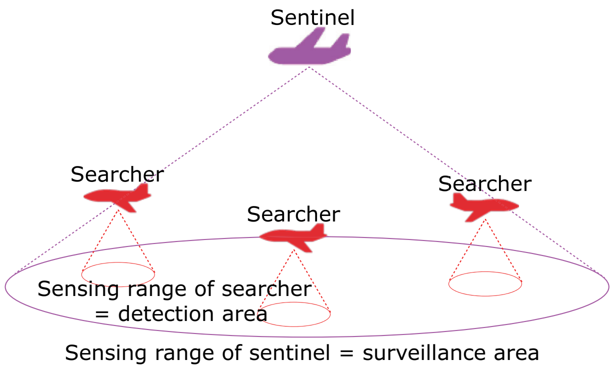

Sentinel
Searcher
Searcher
Searcher
Sensing range of searcher
= detection area
Sensing range of sentinel = surveillance area

Fig. 1. Aerial surveillance system composed of sentinel and searchers

path planning for multiple UAVs in the same area has been proposed [5]. Furthermore, multi-UAV path planning in consideration of the connectivity among the UAVs has been proposed [6]. However, static targets have been assumed to occur randomly at any place, as described in [7].

In this paper, we assume unidentified flight vehicles, UFVs, as surveillance targets. As well as the system illustrated in Fig. 1, one sentinel and multiple searchers are used. Compared to the static targets, searchers are required to approach UFVs in consideration of the movement.

In our previous work [8], the movement of the UFVs was stochastically predicted. According to the movement, optimal approaching strategies have been proposed with the use of the value iteration method. However, the UFVs were assumed to move linearly ignoring searchers. Hence, it is difficult to predict the movement if the UFVs change the actions due to the searchers. Therefore, we apply reinforcement learning to the searchers in consideration of the interaction. In the field of multi-agent reinforcement learning, this issue has been treated as a pursuit game [9]. However, faster pursuers than evaders have been assumed. To our knowledge, a superior evader was assumed to move at the same speed as pursuers [10].

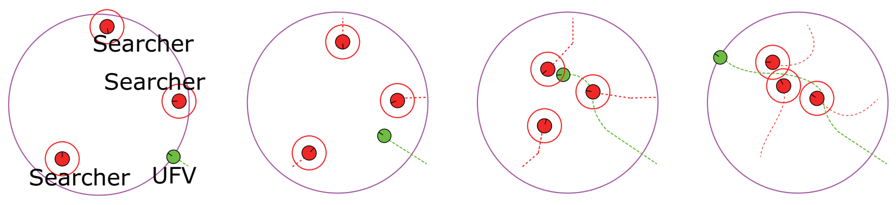
In the aerial surveillance system, on the other hand, we assume the UFVs faster than the searchers. Furthermore, the UFVs are allowed to move while avoiding the searchers through interaction. Therefore, it might be difficult for the searchers to approach and detect such sophisticated UFVs as long as the searchers take actions independently from each other. For this challenge, cooperative actions among the multiple searchers play an important role.

In the reinforcement learning for cooperative surveillance, the searcher state is defined on the basis of a target UFV and other searchers. As for a series of actions, action weights (or policies) in the states are calculated depending on a reward based on the surveillance result. A searcher in a state selects a next action based on the action weight. In this regard, however, the state space is too large to explore. For multi-agent reinforcement learning, a deep Q-network, DQN, has been used for Q-value approximation [11]. In stead of state exploration, therefore, a relationship between the state and action weight is approximately represented by a deep neural network. Through the state-action mapping, action weights are derived as the outputs even for the first input states. In the surveillance system, the searchers operate by repeating the state observation and policy-based decision making.

In the initial learning phase, however, searchers based on random actions seldom detect UFVs. Thus the searchers cannot acquire effective actions to approach the target UFV. Therefore, we further apply transfer learning [12] to this problem. In the initial learning phase, smaller environments with limited state space are provided so that the searchers easily approach and detect UFVs. The learning results are then transferred to the original surveillance environment. Through surveillance simulations, we show the effectiveness of the cooperative actions by the multiple searchers for the UFVs in the aerial surveillance system.

2 Multi-Agent Aerial Surveillance System

Within the surveillance area as illustrated in Fig. 1, the sentinel constantly tracks the positions of searchers and UFVs. The searchers are allowed to communicate with the sentinel. Thus a searcher has information about the target UFV and other searchers via the sentinel. **Fig. 2** illustrates the top view of the surveillance system with three searchers for an UFV as the target. The pink circle indicates the surveillance area. The UFV moves below the searchers.



(a) Searchers vs. incoming UFV (b) Approaching searchers (c) Detection (success) (d) Failure

Fig. 2. Top view of aerial surveillance system with multiple (three) searchers

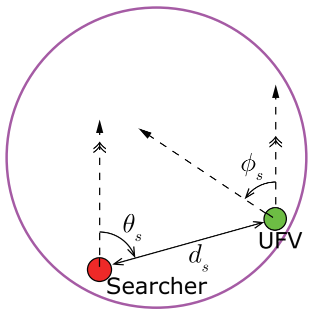
In Fig. 2(a), the incoming UFV is detected by the sentinel. The information is shared with the three searchers. The UFV moves toward a destination located at the edge of the surveillance area. In Fig. 2(b), the searchers approach the UFV by taking actions in each state repeatedly. In Fig. 2(c), eventually, a searcher succeeds in detecting the UFV within the red sensing range. On the other hand, in Fig. 2(d), the UFV reaches the destination without being detected by the searchers. In this case, the searchers fail in detecting the UFV. As well as the searchers, UFVs in the surveillance area are also allowed to have information about the searchers by radar. Thus the UFVs are able to move toward the destinations while avoiding the searchers.

For reinforcement learning, the surveillance process from the detection of the incoming UFV by the sentinel to the success or failure of detection by the searchers is called episode. At the end of each episode, a reward is given to the searchers depending on the surveillance result, success or failure. With each episode, the searchers gradually learn effective actions to approach the UFVs.

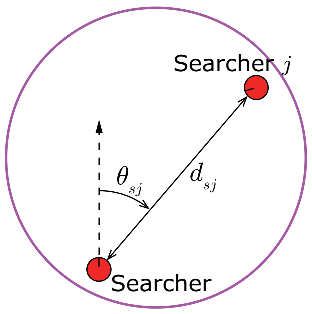
3 Reinforcement Learning for Multiple Searchers

3.1 State Definition

In the aerial surveillance system, faster UFVs move so as to avoid the searchers. In order for the searchers to approach and detect such sophisticated UFVs, cooperative actions among the multiple searchers play an important role. For cooperative surveillance, the searcher state is defined on the basis of the relative position and direction to the target UFV and other searchers as illustrated in **Fig. 3**. The polar coordinate system centered on the host searcher reduces the combination of parameters used for state definition.



(a) State based on UFV



(b) State based on other searcher(s)

Fig. 3. Definition of searcher state in polar coordinate system

The positions and moving directions of the searcher and UFV are represented by x, y, and θ in the 2D plane. As described in section 2, the information is known to both the searchers and UFV. In Fig. 3(a), d_s and θ_s indicate the distance and direction angle of the searcher to the UFV, respectively. A relative direction between the searcher and UFV, ϕ_s, is further included in the state. In Fig. 3(b),

d_{s_j} and θ_{s_j} indicate the distance and direction angle of the searcher to the other searcher, j.

As described above, the searcher state at time t in an episode, s_t, is defined by the combination of $d_{s_{j\in J}}$ and $\theta_{s_{j\in J}}$ in Fig. 3(b) in addition to d_s, θ_s, and ϕ_s in Fig. 3(a). As for the $d_{s_{j\in J}}$ and $\theta_{s_{j\in J}}$, J indicates a set of other searchers j excepting the host searcher i, $i \neq j$. Hence, it is expected that the searchers can learn the approaching actions taking the moving direction of the UFV and cooperative relations with other searchers into account.

3.2 Discretized actions

In a state, s_t, searchers take actions so as to approach the UFV. The searcher action, defined by the directional change $\varDelta\theta$ and moving distance d, is represented as $a_t = \{\varDelta\theta_t^a, d_t^a\}$. Given continuous values of $\varDelta\theta$ and d, the action space becomes too large to explore.

In the system, the searcher action is discretized by dividing into N directions. In doing so, a set of actions in state s_t is represented by $A_t = \{a_t^1, a_t^2, \cdots, a_t^N\}$. A searcher selects an action, a_t, from a set of the discretized actions A_t. For instance, the searcher that selected action a_t^1 rotates $\theta_t^{a^1}$ and moves a distance of $d_t^{a^1}$.

3.3 Rewards based on surveillance results

At the end of each episode, positive or negative rewards are given to the searchers depending on the surveillance result, success (see Fig. 2(c)) or failure (see Fig. 2(d)). If the negative rewards are equally given to the searchers in case of failure, two series of actions of approaching and leaving searchers are both evaluated as ineffective for surveillance. In this case, therefore, the negative rewards depending on the final distance to the UFV are given to the searchers as follows:

$$R = \begin{cases} \quad r \quad \text{(success)} \;\; \text{or} \\ -\frac{d_{fin}}{D} r \; \text{(failure)}, \end{cases} \tag{1}$$

where r is a reward constant, d_{fin} is the distance to the UFV at the end of the episode, and D is a diameter of the surveillance area.

From Eq. (1), if any of the searchers detects the UFV, the positive rewards are equally given to all the searchers. On the other hand, the negative reward increases as the final distance to the UFV increases in case of failure.

4 Surveillance Policy

4.1 Profit sharing with expected value

In order for a searcher in state s_t at time t to take action a_t, action weight $W(s_t, a_t)$ is used as an indicator. As the action weight is increased through reinforcement learning, the searcher tends to select action a_t in state s_t. The action weight in each state is calculated from rewards. As described in section 3.3, the rewards are given to the searchers at the end of the episode depending on the surveillance result.

After the episode, action weights corresponding to a series of actions are calculated. For this purpose, we present profit sharing [13] with expected value, PSw/EV. In addition to the given reward, the action weight is calculated in consideration of the expected value of the reward in each state. After that, the action weight is updated as follows:

$$W(s_t, a_t) \leftarrow W(s_t, a_t) + \gamma^{(\tau - t)}\{R - E(s_t)\}, \tag{2}$$

where γ is a discount rate, τ is an end time of the episode, and $E(s_t)$ denotes the expected value of the reward in state s_t, $E(s_t) = \Sigma_e R(s_t)/N(s_t)$, where $\Sigma_e R(s_t)$ indicates the sum of rewards given in episodes e in which the searcher observed state s_t and $N_e(s_t)$ is the number of episodes e.

During each episode, a series of actions, $a_1, a_2, \cdots$, in states, $s_1, s_2, \cdots$, is stored. After the episode, the action weight at time t, $W(s_t, a_t)$, based on state s_t and action a_t is updated according to the difference between the given reward R and expected value of the reward in state s_t, $E(s_t)$. From Eq. (2), W is increased if R is higher than $E(s_t)$, and vice versa, through the PSw/EV.

4.2 RBFnet for state-action mapping

The searcher state is defined by continuous values as described in section 3.1. Thus in section 4.1, it is impossible to explore the whole state space and calculate $W(S, A)$ for all the state-action pairs, $s \in S$ and $a \in A$. In this paper, therefore, a relationship between the state s and action weight $W(s)$ is formulated. For state-action mapping, a radial basis function network [14], RBFnet, is used.

The RBFnet is based on a deep neural network composed of three, input, hidden, and output layers. In the RBFnet, radial basis functions are applied to the units in the hidden layer. As a result, the relationship between the input state s and output action weight W is represented by the following equation:

$$W(s; d_s, \theta_s, \phi_s, d_{s_{j\in J}}, \theta_{s_{j\in J}}) = \sum_{k=1}^{K} \boldsymbol{w}_k \varphi_k(s), \tag{3}$$

where K is the number of units in the hidden layer. $\boldsymbol{w}_k$ denotes connection weights between k-th unit in the hidden layer and output units. Since the searcher action is discretized into N directions, the connection weights are $\boldsymbol{w}_k = \left[w_k^{a^1}, w_k^{a^2}, \cdots, w_k^{a^N}\right]$ for the N output units. $\varphi_k()$ is the radial basis function based on the following Gaussian kernel:

$$\varphi_k(s) = \exp\left(-\frac{||s - \mu_k||}{\sigma_k^2}\right), \tag{4}$$

where μ_k and σ_k are a central value and variance of the Gaussian kernel in the k-th unit.

Through state-action mapping in Eq. (3), the RBFnet accepts not only an observed state, s, but also the other similar states even if the states are not observed by the searcher. For these input states, the actions weights of $W_{a^1}(s)$ to $W_{a^N}(s)$ are derived from the N output units. **Fig. 4** illustrates an example structure of the RBFnet with three searchers. The input layer is composed of $3 + 2J$ units, i.e., three units for the UFV and $2 \times J$ units for other searchers $j \in J$. Given the three searchers, $J = 2$, the input layer is composed seven units in total. The hidden and output layers are composed of K Gaussian units and N units, respectively.

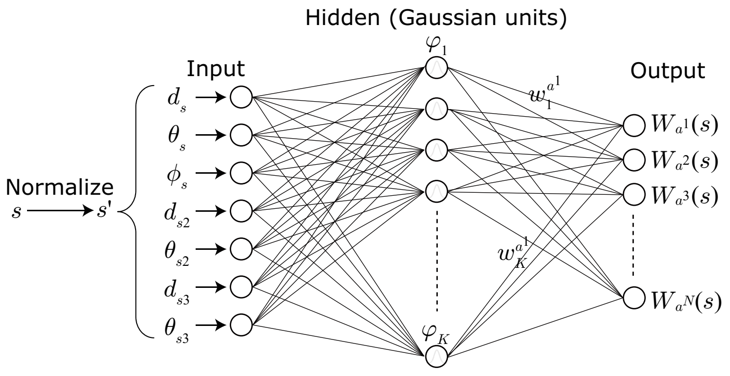


Fig. 4. Structure of RBFnet with three searchers

The input state, s, is first off normalized to s'. Hence, each parameter of d_s, θ_s, ϕ_s, d_{s2}, θ_{s2}, d_{s3}, and θ_{s3} has a value of $[0, 1]$. These values are then fed as the inputs to the RBFnet. After that, φ_1 to φ_K are calculated by Eq. (4) in the K Gaussian units. Through the connection weights of $\boldsymbol{w}$, the action weights, $W_{a^1}(s)$ to $W_{a^N}(s)$, are calculated by Eq. (3) and derived as the outputs.

After each episode, action weights are updated by Eq. (2). For the outputs of $W_{a_t^1}(s_t)$ to $W_{a_t^N}(s_t)$ derived from the RBFnet, only the output unit, $W_{a_t}(s_t)$, corresponding to action a_t taken in state s_t is used as the target. In other words, the connection weights to the output unit in Eq. (3) are updated so that $W_{a_t}(s_t)$ equals to $W(s_t, a_t)$ as follows: $\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k + \eta\delta\varphi_k(s_t)$, where η is a learning rate and δ is a update value, calculated from $\delta = \gamma^{(\tau-t)} \{R - E(s_t)\}$.

4.3 Boltzmann action selection based on action weights

In state s_t at time t, the searcher probabilistically selects an action depending on the action weights of $W_{a_t^1}(s_t)$ to $W_{a_t^N}(s_t)$. For this purpose, Boltzmann action selection [15] is used. The searchers repeat the decision making until the end of the episode. For Boltzmann action selection, each action weight $W_{a_t}(s_t)$ is converted to the following probability, p:

$$p(a_t|s_t) = \frac{\exp\left(\frac{W_{a_t}(s_t)}{T}\right)}{\sum_{a_t^n}^{N} \exp\left(\frac{W_{a_t^n}(s_t)}{T}\right)}, \tag{5}$$

where T is the temperature, which is a time-varying parameter.

From Eq. (5), the probability, p, increases as the action weight, $W_{a_t}(s_t)$, is increased. Therefore, an action with higher weight tends to be selected. In general, parameter T decreases to 0 with time. Eventually, the action weight has a greater impact on Eq. (5). Thus the Boltzmann action selection through state-action mapping in section 4.2 is the surveillance policy for the searchers.

5 Transfer Learning

In the initial learning phase, action weights W are uniformly given to all the states $s \in S$ for all the actions $a \in A$. The surveillance policy allows the searcher to select each action with equal probability as expressed in Eq. (5). In other words, the searchers gradually learn effective actions from the random actions by detecting the target UFV fortuitously.

However, the searchers based on the random actions seldom detect sophisticated UFVs. As long as the searchers fail to detect the UFV, the negative rewards continue to be given to the searchers. Even if a searcher partially takes effective actions, the action weights are not increased in case of failure. As a result, the searchers cannot acquire the effective actions to approach the UFV through reinforcement learning.

This learning problem is due to the large state space. Therefore, we further apply transfer learning to this problem. In transfer learning, the final action weights $W_A(S_{M-1})$ in smaller environment D_{M-1} are used as the initial action weights $W_A(S_M)$ in larger environment D_M. The state space of D_{M-1} and D_M is represented as $S_{M-1} \subseteq S_M$. **Fig. 5** illustrates the process of transfer learning.

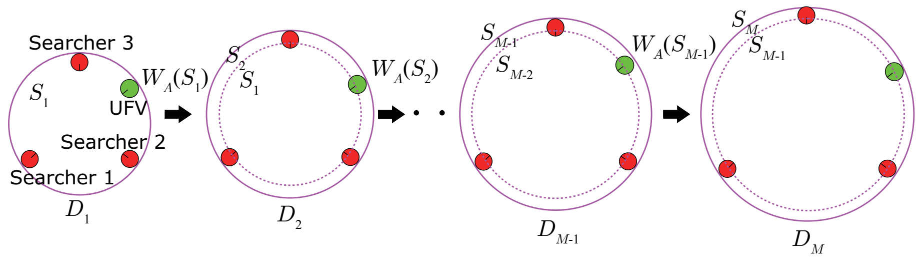


Fig. 5. State space expansion for transfer learning

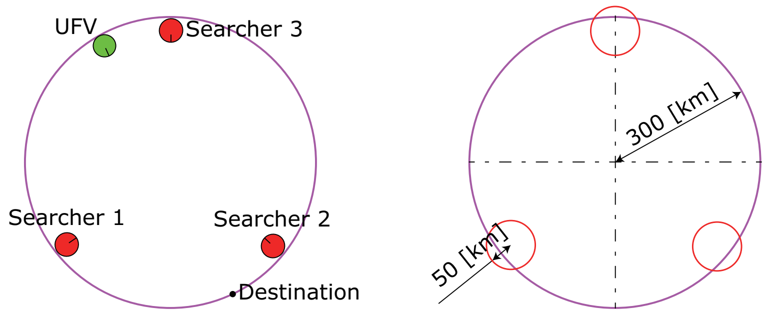
In Environment D_1, the state space S_1 is limited to around the UFV. In the state space, the searchers are already close to the UFV and arranged so as to surround the UFV. In these states, even the searchers based on the random actions can detect the UFV. Moreover, the searchers are allowed to observe the state space and update the action weights sufficiently. After that, the action weights, $W_A(S_1)$, in environment D_1 are transferred to the expanded environment, D_2, and used as the initial action weights. The searchers mainly observe the additional state space, $S_2 - S_1$, and update the action weights $W_A(S_2)$. This process is repeated until the size of the environment reaches the surveillance area D_M.

Through transfer learning, random actions of the searchers in the initial learning phase are confined to the limited state space and additional state space. In the initial limited state space, the searchers detect the UFVs with random actions. Therefore, more positive rewards are given to the searchers. As a result, the learning efficiency is increased, and the searchers in the surveillance area are allowed to learn the effective actions to approach the UFV with episodes.

6 Surveillance Simulation

6.1 Simulation settings

In this simulation experiment, single sentinel and three searchers are used for surveillance. In each episode, a target UFV comes into the surveillance area. **Fig. 6** illustrates the environment of aerial surveillance system. Please note that fixed-wing UAVs different from drones are assumed in the system.



(a) Three searchers vs. target UFV (b) Sensing range

Fig. 6. Surveillance environment

The initial positions of three searchers are as indicated in Fig. 6(a). As for the target UFV, both the incoming position and destination are randomly selected from the edge of the surveillance area. In Fig. 6(b), the sensing ranges of the sentinel and searchers are provided. The radius of the surveillance area is 300

[km]. Within the radius of 50 [km], the searchers detect the UFV, and the detail information is identified.

The fixed-wing searchers move at a velocity of 300 [km/h]. Unlike drones, the searchers cannot control the moving direction rapidly and precisely. Hence, the action is discretized as $A = \{a^1, a^2, a^3, a^4, a^5\}$, by dividing into $N = 5$ directions. Thus the searcher based on action $a = \{\Delta\theta^a, d^a\}$ rotates the direction -90 [deg] by a^1, -45 [deg] by a^2, 0 [deg] by a^3, 45 [deg] by a^4, and 90 [deg] by a^5, and moves a distance of $d^a = 30$ [km], in each step corresponding to 0.1 [h].

The UFV moves at a velocity of 450 [km/h]. In consideration of the interaction with the searchers, the moving direction of the UFV is determined by the resultant force of $\boldsymbol{F}_a$ and $\boldsymbol{F}_r$. $\boldsymbol{F}_a$, the attractive force is generated from the destination as follows:

$$\boldsymbol{F}_a(d_d) = \begin{cases} a_1 d_d + b_1 \; (d_d < d_{th1}) \;\text{ or} \\ \quad b_1 \quad\;\; (d_d \geq d_{th1}), \end{cases} \tag{6}$$

where d_d is a distance to the destination, a_1 and b_1 are coefficients, and d_{th1} is a threshold for the distance. $\boldsymbol{F}_r$, the repulsive force is generated from each of the searchers as follows:

$$\boldsymbol{F}_r = \sum_i \boldsymbol{F}_r(d_i) = \begin{cases} a_2 d_i + b_2 \; (d_i < d_{th2}) \;\text{ or} \\ \quad 0 \quad\;\; (d_i \geq d_{th2}), \end{cases} \tag{7}$$

where d_i is a distance to the i-th searcher, $i = \{1, 2, 3\}$. a_2 and b_2 are coefficients and d_{th2} is a threshold for the distance.

From Eq. (6) and Eq. (7), within a certain distance of d_{th1} and d_{th2}, the attractive force and repulsive force increase as the distance reduces, and affect the UFV. These coefficients and thresholds are given as $a_1 = -6$, $b_1 = 210$, $d_{th1} = 30$, $a_2 = -0.3$, $b_2 = 60$, and $d_{th2} = 200$. As well as a superior evader in [10], the movement model of the UFVs is unknown to the searchers.

6.2 Learning conditions

Under the following three learning conditions, the searchers acquire the effective actions to approach the UFV depending on the combination of the input states and presence or absence of transfer learning:

1. state based on the UFV through non-transfer learning;
2. state based on the UFV and other searchers through non-transfer learning;
3. state based on the UFV and other searchers through transfer learning.

In 1, the searcher state is defined by d_s, θ_s, and ϕ_s. In 2, the searcher state is defined by d_{s2}, θ_{s2}, d_{s3}, and θ_{s3} in addition to d_s, θ_s, and ϕ_s. These normalized parameters are fed as the inputs to the RBFnet in Fig. 4. For these input states, action weights are derived from Eq. (3). In 3, the searchers first operate in a smaller surveillance area with a radius of 200 [km] for transfer learning. After 1000 episodes, the action weights are transferred to the original surveillance area.

In Eq. (1), the reward constant is given as $r = 1$. As for the PSw/EV, the discount rate in Eq. (2) is given as $\gamma = 0.9$. In Eq. (3), the number of Gaussian units in the RBFnet is $K = 10^7$. The variance of the Gaussian kernel in Eq. (4) is $\sigma = 0.142$. For Boltzmann action selection, a constant value is given to the temperature in Eq. (5) as $T = 90$ to reduce hyperparameters.

6.3 Simulation results

Surveillance actions acquired by the searchers through reinforcement learning are evaluated on the basis of the success rate in each 1000 episodes. **Fig. 7** shows the results under the three conditions for 20000 episodes in total.

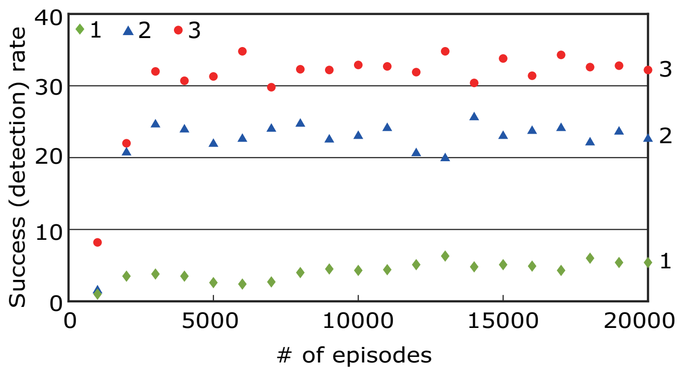
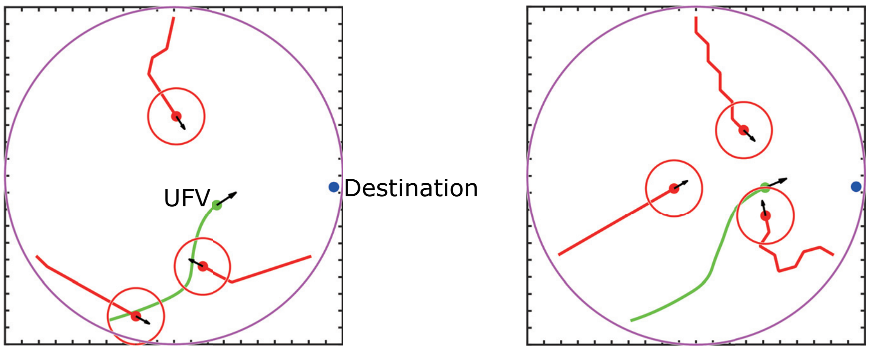


Fig. 7. Transition of success rate: # of detections / 1000 (episodes)

As indicated by $\diamond$ plots, the success rate of surveillance under 1 was not increased with episodes. The final success rate remained at around 5 [%]. On the other hand, the success rate of surveillance under 2 increased to 22.6 [%], i.e., more than quadruple compared to the result of 1, as indicated by $\triangle$ plots. The surveillance actions of the searchers under the conditions, 1 and 2, are compared in an episode. **Fig. 8** shows the approaching trajectories of the searchers for the same target UFV after 20000 episodes. Arrows denote the moving directions of the searchers and UFV at the time.

In Fig. 8(a), the searchers directly approached the UFV independently from each other. As a result of the actions, the searchers failed in detecting the UFV. In Fig. 8(b), on the other hand, the searchers cooperatively approached so as to surround the UFV. As a result of the cooperative actions, the searchers succeeded in detecting the UFV. These results show that the searchers learned the cooperative actions effective for surveillance based on the state defined by the UFV and other searchers.

In Fig. 7, while the final success rate under 2 was 22.6 [%], the initial success rate for the first 1000 episodes was 1.5 [%]. This result was as low as the initial



(a) 1: UFV escaped from searchers (b) 2: detected UFV

Fig. 8. Comparison of trajectories — a series of actions of three searchers

success rate under 1. In contrast, the initial success rate under 3 was increased to 8.2 [%] as indicated by ∘ plots. This result shows that the searchers efficiently learned the effective actions. Because of the increased learning efficiency, the final success rate was also increased to 32.2 [%]. Compared to the result of 2, the surveillance performance was increased more than 10 [%].

As described above, the final success rates were 5.4, 22.6, and 32.2 [%] depending on the learning conditions. For your information, the searchers based on an optimal approaching strategy proposed in [8] resulted in the success rate of 13.4 [%]. As for the remaining failure rates, the UFVs reached the destinations at a rate of 70.1(/94.6), 43.1(/77.4), and 37.4(/67.8) [%], respectively. From the results, the effectiveness of the cooperative actions by the multiple searchers for the UFVs in the aerial surveillance system was shown.

7 Conclusions

In this paper, we proposed an aerial surveillance system composed of single sentinel and multiple searchers. For sophisticated UFVs, the searchers successfully acquired effective actions through reinforcement learning. In the surveillance simulations, the searchers approached the target UFV in cooperation with each other. As a result, the success rate of surveillance was increased to 22.6 [%] from 5.4 [%]. Furthermore, transfer learning increased the initial success rate from 1.5 [%] to 8.2 [%]. Because of the increased learning efficiency, the final success rate was also increased to 32.2 [%], and the system yielded the best surveillance performance. From the results, the effectiveness of the cooperative actions by the multiple searchers for the UFVs in the aerial surveillance system was shown.

In this experiment, we used a computer with Intel Core i7-6950X (CPU), NVIDIA TITAN RTX (GPU), and 64G bytes of memory. As for the computational cost, condition 1 (three parameters for state definition) spent about

24 hours and conditions 2 and 3 (seven parameters for state definition) spent about four days in the simulations. In order for the surveillance system to further increase the success rates, more searchers are required. However, the computational cost increases with the number of searchers. For the curse of dimensionality, transfer learning would be a possible solution. In future works, therefore, we will expand the state space little by little in applying transfer learning.

References

1. Cabreira, T. M., Brisolara, L. B., and Ferreira Jr., P. R.: Survey on Coverage Path Planning with Unmanned Aerial Vehicles, *Drones*, Vol. 3, No. 4, 2019.
2. Zhou, X., Yi, Z., Liu, Y., Huang, K., and Huang. H.: Survey on Path and View Planning for UAVs, *Virtual Reality & Intelligent Hardware*, Vol. 2, No. 1, pp. 56–69, 2020.
3. Basilico, N. and Carpin, S.: Deploying Teams of Heterogeneous UAVs in Cooperative Two-Level Surveillance Missions, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.610–615, 2015.
4. Maza, I. and Ollero, A.: Multiple UAV Cooperative Searching Operation Using Polygon Area Decomposition and Efficient Coverage Algorithms, *Distributed Autonomous Robotic Systems*, pp. 221–230, 2007.
5. Flint, M., Polycarpou, M., and Fernandez-Gaucherand, E.: Cooperative Control for Multiple Autonomous UAV's Searching for Targets, *IEEE Conference on Decision and Control*, pp. 2823–2828, 2002.
6. Hayat, S., Yanmaz, E., Bettstetter, C., and Brown, T. X.: Multi-Objective Drone Path Planning for Search and Rescue with Quality-of-Service Requirements, *Autonomous Robots*, Vol. 44, pp. 1183–1198, 2020.
7. Basilico, N., Chung, S., and Carpin, S.: Distributed Online Patrolling with Multi-Agent Teams of Sentinels and Searchers, *Distributed Autonomous Robotic Systems*, pp. 3–16, 2016.
8. Hoshino, S., Katsumoto, T., Saitoh, S., Itabashi, N., Kabayama, T., and Kono, M.: Aerial Surveillance System with Multiple Agents for Moving Targets, *SICE Annual Conference*, pp. 427–432, 2020.
9. Arai, S. and Sycara, K.: Effective Learning Approach for Planning and Scheduling in Multi-Agent Domain, *International Conference on Simulation of Adaptive Behavior*, pp. 507–516, 2000.
10. Wang, Y., Dong, L., and Sun, C.: Cooperative Control for Multi-Player Pursuit-Evasion Games with Reinforcement Learning, *Neurocomputing*, Vol. 412, 101–114, 2020.
11. Omidshafiei, S., Pazis, J., Amato, N., How, J. P., and Vian, J.: Deep Decentralized Multi-Task Multi-Agent Reinforcement Learning under Partial Observability, *International Conference on Machine Learning*, pp. 2681–2690, 2017.
12. Taylor, M. E. and Stone, P.: Transfer Learning for Reinforcement Learning Domains: A Survey, *Journal of Machine Learning Research*, Vol. 10, No. 5, pp. 1633–1685, 2009.
13. Grefenstette, J. J.: Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, *Machine Learning*, Vol. 3, pp. 225–245, 1988.
14. Poggio, T. and Girosi, F.: Networks for Approximation and Learning, *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1481–1497, 1990.
15. Sutton, R. S. and Barto, A. G.: Reinforcement learning, MIT Press, pp. 32–33, 1998.