Clutter Assessment for an Autonomous Multi-Agent Search Mission

John R. Cooper* and B. Danette Allen†
NASA Langley Research Center, Hampton, VA, 23681

This paper presents a method for evaluating the amount of clutter in a region where autonomous vehicles in a multi-agent system must operate based on LIDAR point cloud measurements. The point cloud is used to generate an occupancy grid which is then projected onto a 2D plane of vehicle motion, constituting an image. A series of Gaussian radial basis functions (GRBFs) is created, each centered at an occupied pixel in a 2D image, and summed together to form the *clutter field*. The clutter field is a representation of the density and permeability of the space at each coordinate. The clutter field is then approximated such that iso-clutter contours are simple geometric objects so that intelligent machine assets can easily query the distance between them and any given point in an environment. In this way, agents are able to determine whether to enter into or steer away from areas of interest. Each vehicle has a clutter threshold representing the clutter value of the space in which it can safely maneuver. The iso-clutter contour corresponding to a vehicle's clutter threshold is treated as the boundary of an obstacle to be avoided. A simulation is presented where a multi-agent system is tasked with persistent observation of a cluttered area. Each vehicle in the simulation has a different clutter threshold. The vehicles use a potential field-based guidance algorithm, and an allocation of vehicles to specific regions of the space emerges.

I. Introduction

This paper presents a method for quantifying clutter that could impede vehicle motion. It is shown how the clutter values can be used in guidance of a multi-agent system to autonomously allocate vehicles in a persistent observation or search and rescue (SAR) scenario. Clutter assessment was a part of NASA's Autonomy Teaming and TRAjectories for Complex Trusted Operational Reliability (ATTRACTOR) project. The objective of ATTRACTOR was to build a foundation for trustworthiness and trust verification in multi-asset human-machine teaming in the operational context of mission-critical planning and execution. To focus the project, the Design Reference Mission (DRM) for ATTRACTOR is a multi-asset search and rescue mission governed by a team consisting of human and machine operators. A specific instance of SAR Under the tree Canopy (SARUC) was the focus of a collaborative partnership between NASA and MIT [1] that was awarded the 2019 AUVSI XCELLENCE Humanitarian Award [2] for "locating humans and materials lost in challenging and cluttered forest areas." Figure 1 shows experimental flight results from the SARUC mission.

Unmanned Aerial Systems (UAS) show great promise for increasing the efficiency, pace, and efficacy of tasks and operations performed for the greater good of society but are, more often than not, operated as first-person extensions of a human operator. For autonomous SARUC systems, navigation through cluttered environments is an important function. Just as humans would do, UAS should be able to assess whether or not they are capable of proceeding with their assigned mission and calculating the probability of success via a confidence metric. Part of this confidence metric [3] is characterizing the "clutter" ahead on their search trajectory and evaluating whether or not to fly into or around a cluttered region. This demands measurements of density, permeability, and a self-assessment of an ownship's size, dynamics, and survivability.

Distributed control of multi-agent systems has been an area of continued research interest for many years [4–6]. This paper specifically addresses a persistent observation/SAR scenario, but distributed control of multiple aerial vehicles can be applied to many other domains including cartography, reconnaissance, and package delivery. Some of the earliest work on distributed multi-agent control involved the consensus problem [7, 8], where each agent updates its own estimate of some value based on its interactions with other agents in the system. Consensus is achieved when all agents' estimates converge to the same value. In [9–13], consensus algorithms were developed for various information flow constraints. A more complex problem in multi-agent control is formation control, where rather than converge to a single value,

*Research Aerospace Technologist, Navigation, Guidance, and Control Systems, Autonomous Integrated Systems Research Branch, MS 233.
†NASA Senior Technologist for Intelligent Flight Systems, MS 233, and AIAA Associate Fellow.

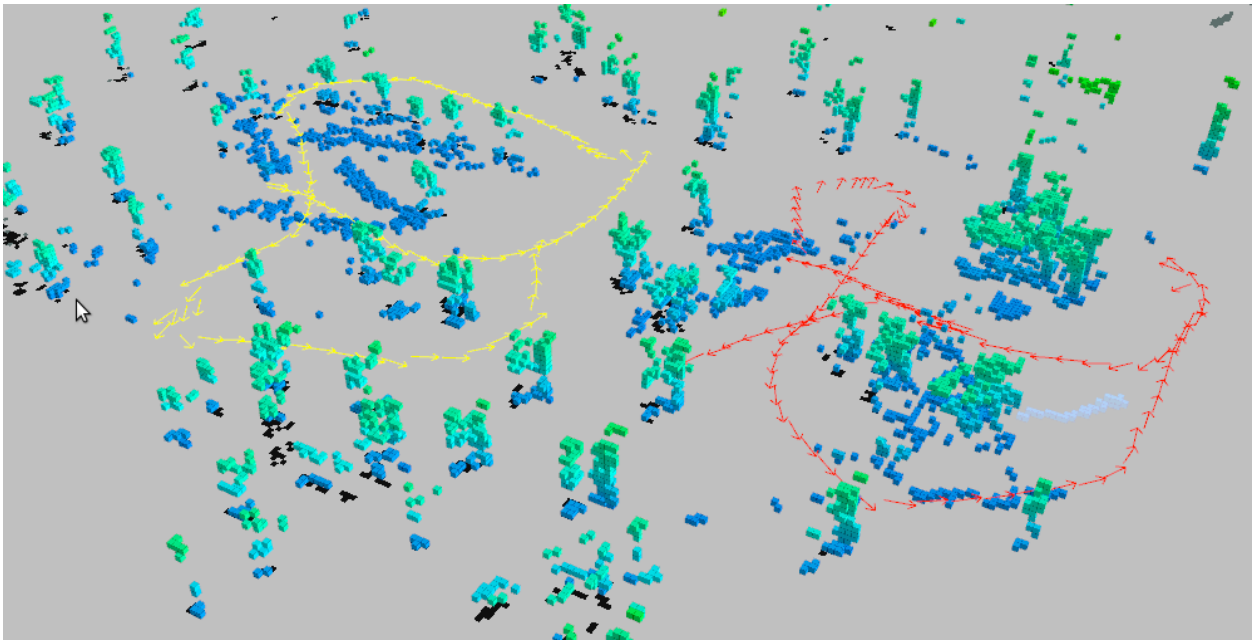


Fig. 1 SARUC collaborative exploration of the NASA Langley Research Center forest by two vehicles [1]

each agent's state may converge to a unique value to optimize a global objective function [14–16]. Formation control finds applications in formation flying, cooperative transportation, sensor networks, combat intelligence, surveillance, and reconnaissance [17–19]. Other extensions to multi-agent control include game-theoretic approaches [20–24] and adaptive methods [25–27].

The paper is organized as follows: section II describes how the clutter field is defined and evaluated, section III provides the guidance algorithm which uses the clutter field to autonomously allocate agents, section IV gives a simulation example, and conclusions are drawn in section V.

II. Clutter Field Definition

The clutter field is defined over the space of vehicle motion such that every coordinate in that space has an associated clutter value. The clutter value can be thought of as the probability of an obstacle existing at this location. A 2D plane of vehicle motion is considered in this paper for clarity, but the clutter field can be defined for higher dimensional spaces as well.

Given an occupancy grid projected over the plane of the vehicles' motion, we define the clutter field using a summation of Gaussian radial basis functions (GRBFs) centered at each occupied pixel,

$$Z(\xi) \triangleq A \sum_{i=1}^{N} \exp\left(-\frac{||\xi - \mu_i||}{\delta_\mu}\right) \tag{1}$$

where Z is the clutter value, ξ is the coordinate in the plane at which the field is being evaluated, A is a scaling parameter, N is the number of occupied pixels, μ_i is the center of occupied pixel i, and δ_μ is the GRBF width. Such an occupancy grid can be obtained for example from LIDAR measurements. Contours of constant Z are referred to as *iso-clutter contours*. To use the clutter field in a guidance law, the point on each iso-clutter contour corresponding to each vehicle's *clutter threshold* must be evaluated. The clutter threshold is the clutter field value above which the vehicle cannot safely maneuver to avoid obstacles.

The surface in Eq. (1) can quickly become complex with increasing N. Therefore, a simpler surface is fit to the actual clutter field such that the minimum distance to approximate iso-clutter contours can be efficiently evaluated. The clutter field is segmented based on local maxima, and a single multivariate Gaussian function is fit to each region. The iso-clutter contours of a single multivariate Gaussian function are ellipses, the minimum distance to which can be calculated quickly.

The clutter field is segmented using a maximum filter with a dilation parameter, β. To implement the maximum filter, the clutter field is represented as an image where $Z(\xi)$ is the pixel intensity at pixel position ξ. The filter dilates the image and merges local maxima in Z which are closer than the distance, β. Each segment is the set of points that are closest to each local maximum. To perform this computation, $Z(\xi)$ must first be evaluated at each pixel coordinate.

After segmenting the field, points from each segment are randomly sampled with a probability equal to the actual clutter value at that position. This sampling can be performed in practice with the NumPy function `numpy.random.choice` [28]. The sample mean and covariance are used to fit the multivariate Gaussian for that segment. Finally, a scaling parameter is calculated to fit the multivariate Gaussian for segment j to the true clutter field according to

$$\hat{A}_j(\beta) = \frac{\sum_{i=1}^{N_s} Z(\xi_i)}{\sum_{i=1}^{N_s} \exp\left(-\frac{1}{2}(\xi_i - \hat{\mu}_j(\beta))^\top \hat{\Sigma}_j(\beta)^{-1}(\xi_i - \hat{\mu}_j(\beta))\right)} \tag{2}$$

where $\hat{A}_j$ is the scaling parameter for segment j, N_s is the number of samples, $\hat{\Sigma}_j$ is the sample covariance for segment j, and $\hat{\mu}_j$ is the sample mean for segment j. Note that the sample mean and covariance, and thus $\hat{A}_j$, depend on the dilation parameter, β, used to perform the segmentation. The approximate clutter threshold for segment j is then

$$\hat{Z}_j(\xi, \beta) = \hat{A}_j(\beta) \exp\left(-\frac{1}{2}(\xi - \hat{\mu}_j(\beta))^\top \hat{\Sigma}_j(\beta)^{-1}(\xi - \hat{\mu}_j(\beta))\right) \tag{3}$$

and the total approximate clutter field is given by

$$\hat{Z}(\xi, \beta) = \hat{Z}_j(\xi, \beta) : j = \arg\min_j ||\xi - c_j(\beta)|| \tag{4}$$

where c_j is the centroid of region j.

The dilation parameter is chosen via a brute force minimization of the error function, $\sum_{i=1}^{N_e} ||Z(\xi_i) - \hat{Z}(\xi_i, \beta)||$ where N_e is the number of points at which the clutter field is evaluated. N_a values for the dilation parameter are tested and the result with smallest error is used to approximate the clutter field.

This process is described using psuedocode in algorithm 1.

Algorithm 1 Clutter field approximation algorithm

Receive occupancy grid from an external perception system
Choose N_e values of ξ at which to evaluate the clutter field to create an image to perform segmentation
Evaluate $Z(\xi)$ according to Eq. (1) at each of the N_e values of ξ
for $\beta = 1, 2, \ldots, N_a$ **do**
    Find local maxima in the clutter field using the maximum filter with dilation parameter β
    $n_s \leftarrow$ number of local maxima returned by maximum filter
    for $j = 1, 2, \ldots, n_s$ **do**
        Randomly sample N_s values of ξ in segment j with probability $Z(\xi)$
        Find the sample mean and covariance of the sampled values of ξ
        Calculate $\hat{A}_j(\beta)$ according to Eq. (2)
    Evaluate the approximate clutter field, $\hat{Z}(\xi, \beta)$, according to Eq. (4) at each of the N_e values of ξ
$\beta \leftarrow \arg\min_\beta \sum_{i=1}^{N_e} ||Z(\xi_i) - \hat{Z}(\xi_i, \beta)||$
return $\hat{Z}(\xi, \beta)$

III. Guidance Algorithm

A. Potential Field Construction

The guidance algorithm used to autonomously allocate vehicles is a modification of the one first presented in [29]. The multi-agent guidance is has three objectives: efficiently search the space, avoid collisions with other vehicles, and avoid entering regions where the clutter value exceeds the vehicle clutter threshold. To construct a guidance law that balances these objectives, we define a potential field for each vehicle with terms for each objective. The total potential

of vehicle i is $V_i = V_{s_i} + V_{c_i} + V_{\tau_i}$ where V_{s_i} is the search component, V_{c_i} is the collision avoidance component, and V_{τ_i} is the clutter avoidance component.

The component of the potential field that drives each vehicle to efficiently search the space is defined using a set of GRBFs. The GRBF centers are distributed over the search area. They can be equally distributed, or they can be used to represent specific locations that must be continually observed. The search component of the potential of vehicle i is then

$$V_{s_i} = -\frac{W_s}{2} \sum_{j=1}^{m} \gamma_{ij} r_j \exp\left(-\frac{||\xi_i - p_j||^2}{\delta}\right) \tag{5}$$

where $W_s > 0$ is the weight prescribed to the searching goal, j is an index for each GRBF center, m is the number of GRBFs in the search area, γ_{ij} is a boolean which is true if $\hat{Z}(p_j, \beta) < \tau_i$ where τ_i is the clutter threshold of vehicle i, r_j is a time-varying reward for being within measurement range of the center of GRBF j, p_j is the position of the center of GRBF j, ξ_i is the position of vehicle i, and $\delta > 0$ is the GRBF width. δ is a tuning parameter that should be chosen such that there is some overlap in the GRBFs. If δ is too small, the potential field gradient will be small between GRBFs, which will cause ineffective guidance commands. If δ is too large, the local minimum at each GRBF center will be lost, and vehicles will be guided to a minimum at some location between GRBFs. Note that GRBFs centered at a location too cluttered for vehicle i do not affect vehicle i's potential.

The purpose of the reward variables, r_j is to increase the weights in the overall potential for areas of the search space which haven't been observed recently and decrease the weights for locations which have been observed recently. Each r_j is updated according to

$$\dot{r}_j = K_{r+}(1 - \alpha_j) - K_{r-}\alpha_j r_j \tag{6}$$

where $K_{r+}, K_{r-} > 0$ are reward rate gains, and α_j is given by

$$\alpha_j = \begin{cases} 0 & \text{if } \min_k ||\xi_k - p_j|| > \mu \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

where $k = 1, \ldots, n$ is the vehicle index and $\mu > 0$ is the vehicle measurement range. Equation (7) means that α_j is 0 if no vehicle is within measurement range of GRBF j and 1 otherwise. In conjunction with Eq. (6), this means that the reward for visiting p_j increases at a rate of K_{r+} until a vehicle goes there. Once any vehicle is within measurement range of p_j, this reward begins to decrease as a first-order linear system with a steady-state value of 0 and bandwidth K_{r-}. Note that communication amongst all agents is assumed.

The component of the potential field that drives each vehicle away from others is defined using the inverse of the distance between agents and is given by

$$V_{c_i} = \frac{W_c}{2} \sum_{k=1, k\neq i}^{n} ||\xi_i - \xi_k||^{-2} \tag{8}$$

where $W_c > 0$ is the weight prescribed to the collision avoidance goal, and n is the number of vehicles involved in the search. As the distance between agents approaches 0, V_{c_i} approaches infinity. Thus, the collision avoidance component will always dominate the potential when two vehicles get close enough to each other. The magnitude of W_c controls the distance at which the collision avoidance term begins to dominate the search term.

The collision avoidance term of the potential field may also be viewed as a deterrent for each agent against duplicating the efforts of others. As a result of Eq. (8), potential will be higher in areas which other vehicles are closer to. In seeking low potential, vehicles will be driven away from areas of the search space that can be more efficiently searched by others.

Finally, the term for avoiding areas that exceed the vehicle's clutter threshold is

$$V_{\tau_i} = \frac{W_\tau}{2} \sum_{l=1}^{n_s} d_{il\tau}^{-2} \tag{9}$$

where $W_\tau > 0$ is the weight for avoiding cluttered regions, l is an index for each segment of the approximate clutter field resulting from the segmentation step of algorithm 1, and $d_{il\tau}$ is the distance from vehicle i to the iso-clutter contour of segment l with clutter value τ. The iso-clutter contour is found by setting $\hat{Z}(\xi, \beta) = \tau_i$ and solving for ξ. Let the set of ξ that satisfies this condition in segment l be $\xi_{l\tau}$. Next, an objective function, $E_l(\xi_{l\tau})$ is defined

$$E_l(\xi_{l\tau}) = (\xi_i - \xi_{l\tau})^\top (\xi_i - \xi_{l\tau}) \tag{10}$$

Then

$$d_{il\tau} = ||\xi_i - \arg\min(E_l(\xi_{l\tau}))|| \tag{11}$$

B. Velocity Command

Each vehicle's overall goal is to minimize its own potential, V_i. The guidance algorithm generates velocity commands in the direction of steepest descent of the potential function with magnitude proportional to the gradient and saturated by a maximum speed, v_{max}.

The guidance algorithm operates similarly to stochastic gradient descent (SGD) [30], which is widely used to efficiently optimize cost functions for fitting models to large amounts of data. In SGD, each gradient descent step uses a randomly sampled subset of the data to be fit. Thus the actual function being minimized changes by a small amount each step. Despite this change, the gradient direction from the previous step is often very close to optimal for the current step. In the guidance algorithm presented here, gradient descent of the potential field happens continuously, and the potential field also changes continuously. Over the course of one infinitesimal time-step, the change in the potential field is small, and thus the commanded direction of travel for any vehicle at the previous time-instant is close to the direction that would have been prescribed by the current gradient. By continuously making these small adjustments, vehicle trajectories over the course of a mission remain close to optimal.

The velocity command is

$$v_i = \begin{cases} -K\frac{\partial V_i}{\partial \xi_i} & \text{if } \left\| K\frac{\partial V_i}{\partial \xi_i} \right\| \leq v_{max} \\ -v_{max}\frac{\frac{\partial V_i}{\partial \xi_i}}{\left\|\frac{\partial V_i}{\partial \xi_i}\right\|} & \text{otherwise} \end{cases} \tag{12}$$

where $K > 0$ is a control gain. The partial derivative of potential with respect to the agent's own position is used instead of considering all agents in the system because each agent only has authority over its own position and velocity. Thus, the guidance algorithm is decentralized.

To calculate the partial derivative in Eq. (12), we express V_i in terms of Eq. (5), Eq. (8), and Eq. (9) as

$$V_i = -\frac{W_s}{2}\sum_{j=1}^{m} r_j \exp\left(-\frac{||\xi_i - p_j||^2}{\delta}\right) + \frac{W_c}{2}\sum_{k=1,k\neq i}^{n} ||\xi_i - \xi_k||^{-2} + \frac{W_\tau}{2}\sum_{l=1}^{n_s} d_{il\tau}^{-2} \tag{13}$$

Then the partial derivative with respect to ξ_i is

$$\frac{\partial V_i}{\partial \xi_i} = \frac{W_s}{\delta}\sum_{j=1}^{m} r_j \exp\left(-\frac{||\xi_i - p_j||^2}{\delta}\right)(\xi_i - p_j) - W_c\sum_{k=1,k\neq i}^{n}\frac{\xi_i - \xi_k}{||\xi_i - \xi_k||^4} - W_\tau\sum_{l=1}^{n_s}\frac{\xi_i - \arg\min(E_l(\xi_{l\tau}))}{||\xi_i - \arg\min(E_l(\xi_{l\tau}))||^4} \tag{14}$$

The velocity commands can be computed by a global agent and then sent to each agent, or they can be computed on board the individual agents. Note that communication amongst agents is required since the velocity command requires knowledge of the state of each observation point and each other vehicle.

IV. Simulation Example

A. Clutter Field Approximation

This section provides a demonstration of algorithm 1. An example 2D projection of a LIDAR point cloud is shown in figure 2. The clutter field resulting from Eq. (1) with $\delta_\mu = 50$ applied to the point cloud projection in figure 2 is shown in figure 3a. The clutter field is evaluated at $N_e = 289$ points. The points are the vertices of a 64×64 grid with 4 unit spacing on each axis. Note that the iso-clutter contours have complex geometry. This prevents quick and efficient proximity queries.

The next step is to separate the field into regions that correspond to local maxima. The simulation used $N_a = 32$, and the β value is 1. Because the local maxima of the original clutter field are sufficiently far from each other, the maximum filter can preserve all the detail of the original field while still finding a good fit. If the shape of the original field was more complex, a higher β would be needed to avoid overfitting in the segmentation of the field. Selection of β is performed automatically at line 12 of algorithm 1. Figure 4 shows the Voronoi diagram resulting from the three local maxima. The Voronoi regions will each have a separate multivariate Gaussian function fit to the field within.

Figure 5a shows the resulting approximate clutter field from fitting a multivariate Gaussian function to the original surface in each of the Voronoi regions using $N_s = 1000$. Note that the iso-clutter contours are now ellipses. Efficient proximity queries can be performed.

Finally, the clutter field approximation is validated by comparing it to the original surface. Figure 6 shows the surface resulting from the approximation error. The largest error is small compared to the clutter values.

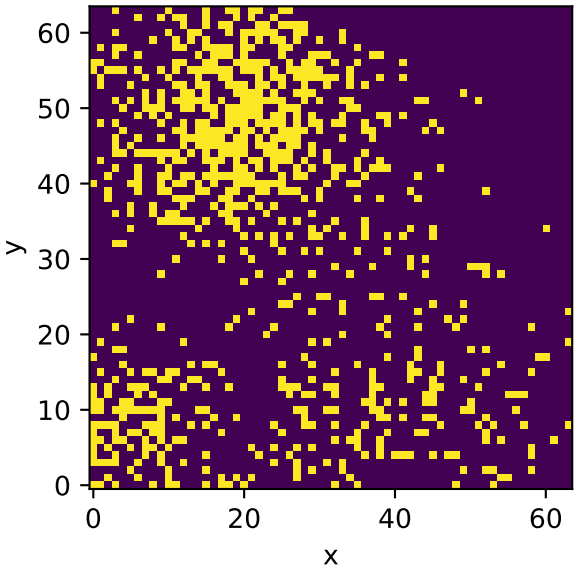
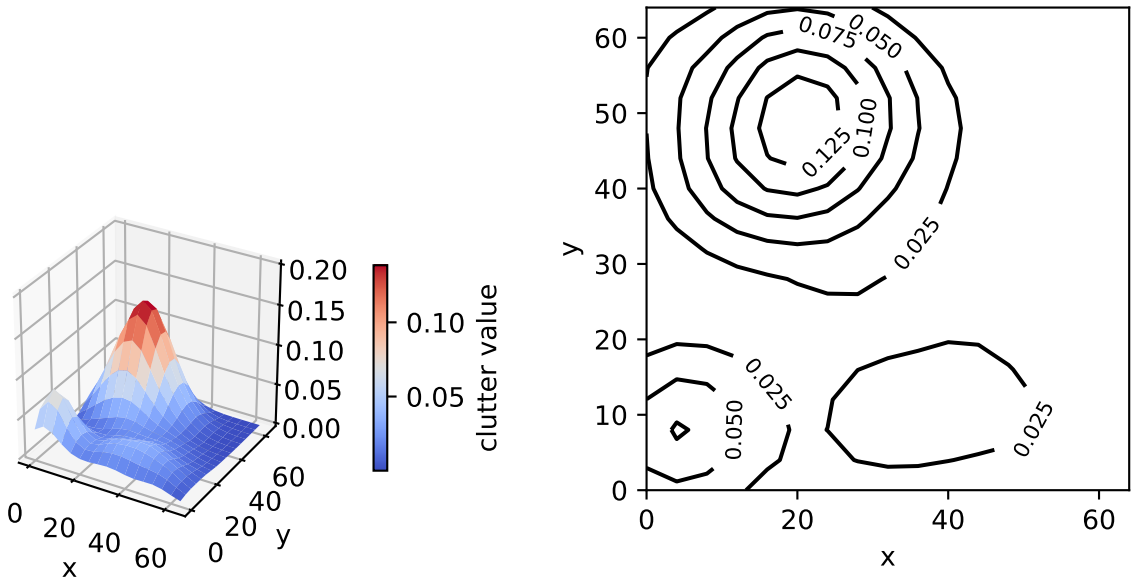


Fig. 2 2D projection of a point cloud. Yellow represents occupied pixels, and indigo represents unoccupied pixels



(a) 3D surface representing the clutter field **(b) Iso-clutter contours**

Fig. 3 Clutter field

B. Guidance Results

This section presents the results of a simulation of three vehicles, each with different clutter thresholds. The goal is for the vehicles to persistently observe the space over which the clutter field in section IV.A is defined. The GRBF centers are defined as the vertices of a grid with 8-unit spacing in both axes, resulting in 64 observation points. Each vehicle has a measurement radius, μ, of 8 and a maximum speed, v_{max}, of 8. The three vehicles have clutter thresholds, $\tau_1 = 0.03$, $\tau_2 = 0.12$, and $\tau_3 = 1.0$. All other simulation parameters are provided in table 1.

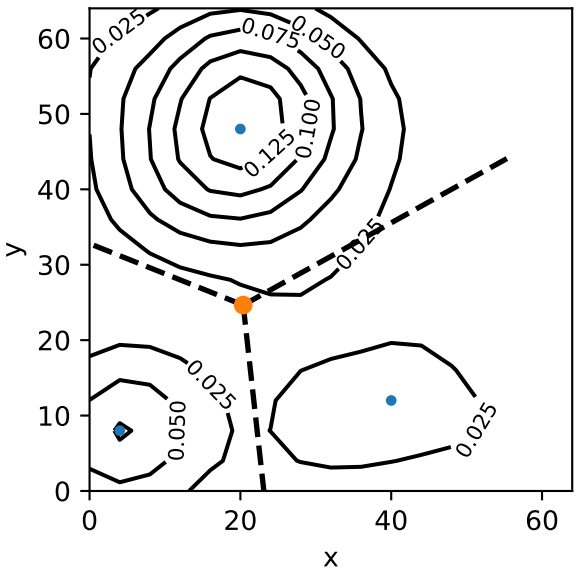
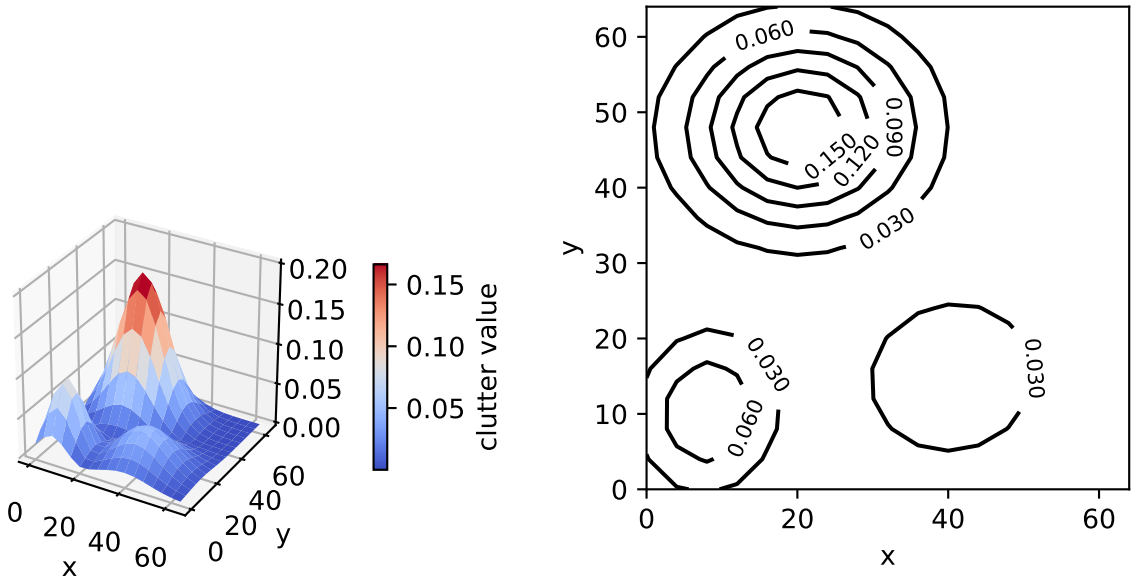


Fig. 4 Voronoi diagram of the three local maxima overlaid with the iso-clutter contours.



(a) 3D surface representing the approximate clutter field **(b) Approximate iso-clutter contours**

Fig. 5 Approximate clutter field

Figure 7 shows the trajectories of all vehicles over the search area. The GRBF centers are represented as gray circles. The iso-clutter contours of the approximate clutter field are also overlaid. The vehicles self-allocate to cover the regions of the space that they are best suited for, and no vehicles cross the iso-clutter contours of their clutter thresholds. Agent 1 has the lowest clutter threshold, and thus it self-allocates to the eastern region of the space. Agent 2 can visit all but the most cluttered regions, and it self-allocates to the western region. Agent 3 can visit anywhere in the space, and it self-allocates to patrol the center region which includes the most cluttered area.

Figure 8 shows how the potential values and reward values change throughout the simulation. Since neither the

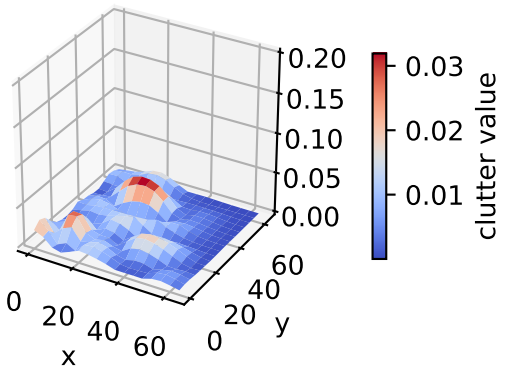


Fig. 6 Error surface between the original clutter field and its approximation

Table 1 Example simulation parameters

Parameter	Value
time-step	1/60
r_0	15
K_{r+}	1
K_{r-}	1
W_c	50,000
W_s	100
W_τ	50
δ	128
K	5

potential values nor the GRBF rewards grow unbounded, it is inferred that the search area is well covered for the duration of the simulation.

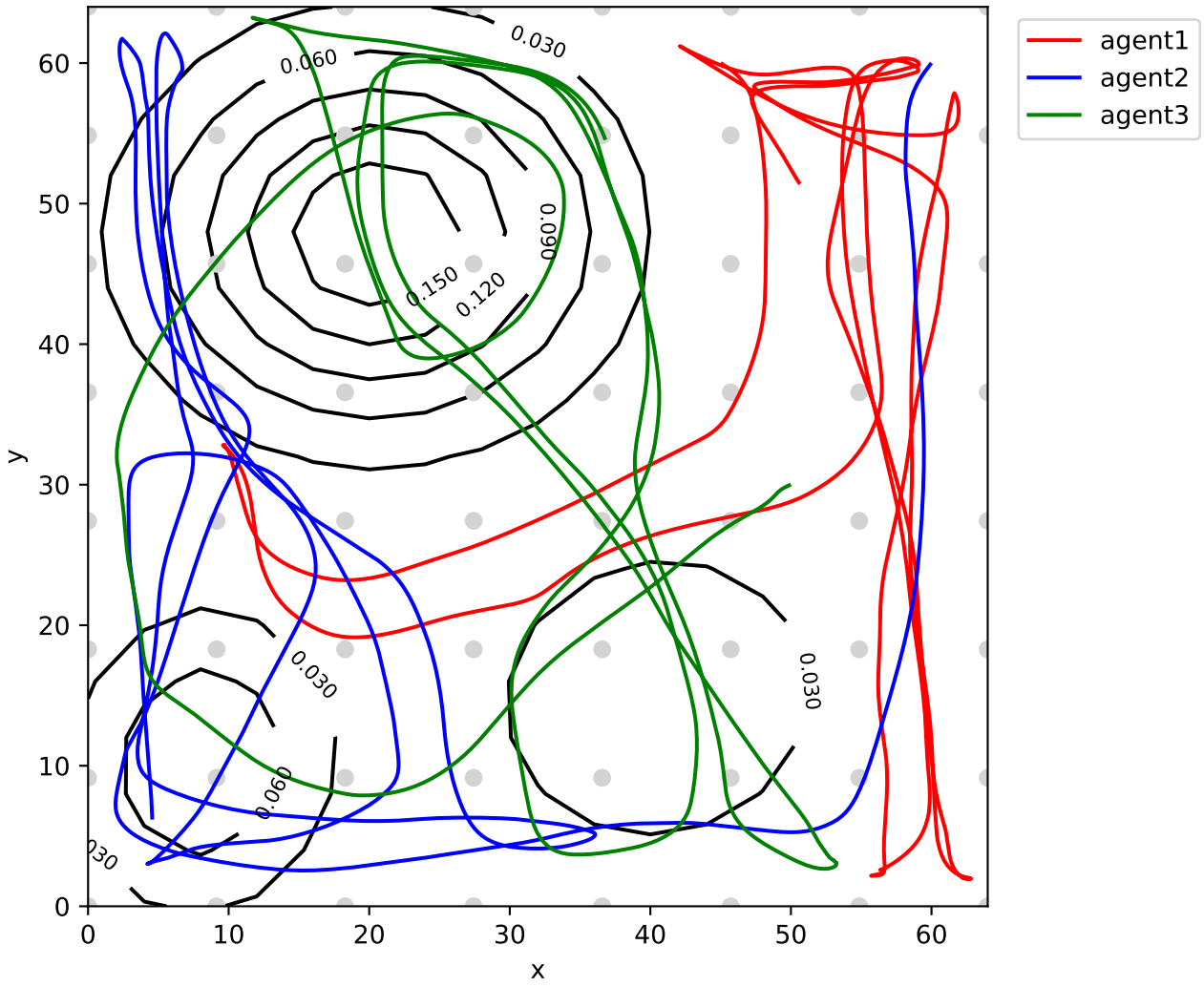
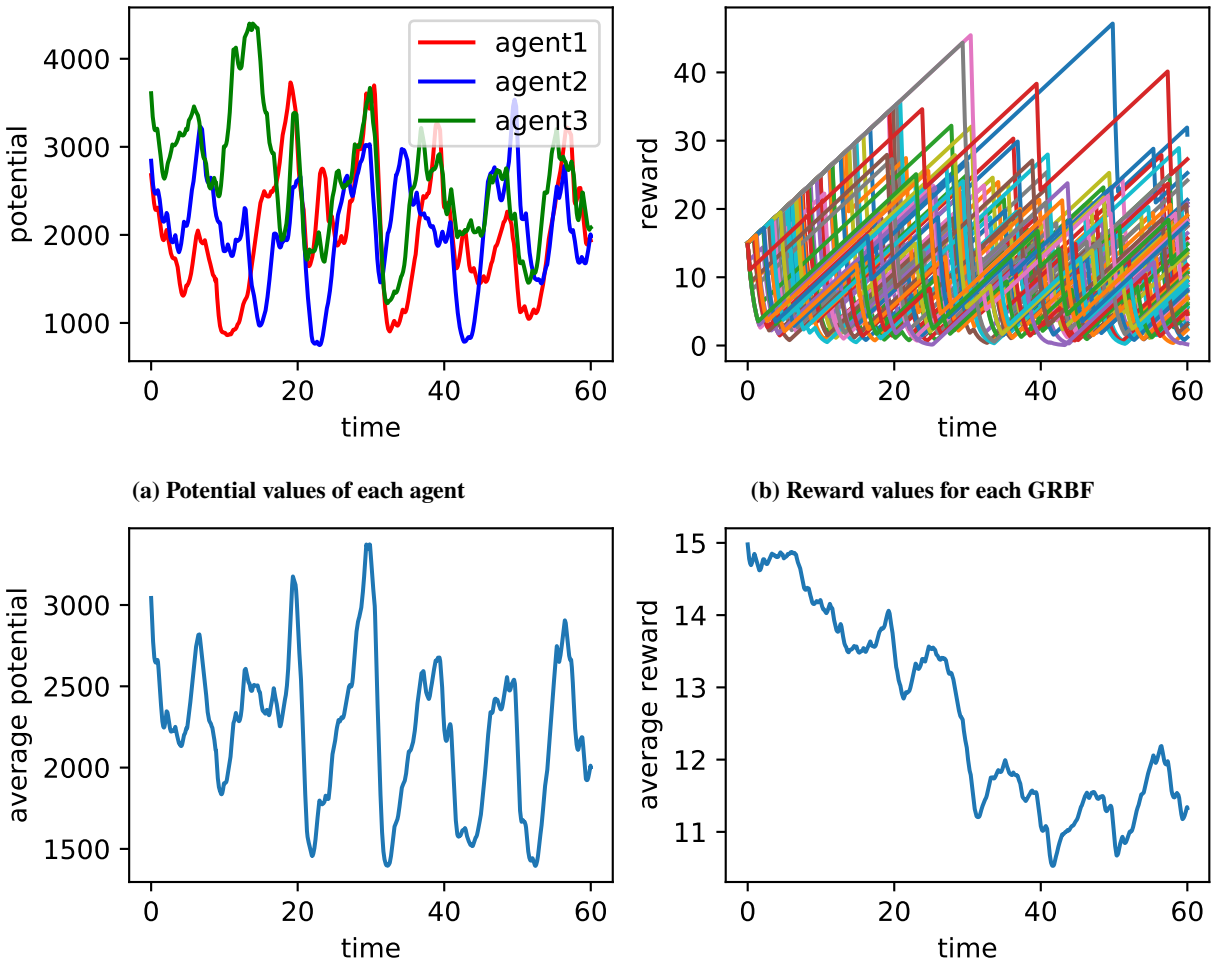


Fig. 7 Vehicle trajectories through clutter field



(a) Potential values of each agent

(b) Reward values for each GRBF

(c) Average potential value for all agents

(d) Average reward value for all GRBFs

Fig. 8 Simulation results

V. Conclusion

This paper presented a method to quantify and assess the density and permeability of cluttered regions in the operational area of autonomous vehicle agents. Additionally, it was shown how the clutter assessment can be used in a potential field-based guidance algorithm for persistent observation by a multi-agent system. It was shown through simulation how the clutter field leads to emergence of self-allocating behavior of the agents. Vehicles remain mostly in the regions best suited to the capabilities defined by their clutter thresholds even when no boundaries other than the iso-clutter contours corresponding to individual clutter thresholds are defined. Note that these iso-clutter contours are "keep-out" zones, not "keep-in" zones.

It should be noted that clutter fields combined with clutter thresholds inform where agents should travel on a macro level and do not address physical collision avoidance with objects in the cluttered regions. A complementary lower-level system such as [31, 32] is needed to generate detailed navigation for conflict-free trajectories. The Clutter Assessment guidance algorithm is a high-level system providing mission direction for the vehicles to travel.

Acknowledgments

This work was performed as part of the Autonomy Teaming & TRAjectories for Complex Trusted Operational Reliability (ATTRACTOR) research thrust under the Convergent Aeronautics Solutions (CAS) project of NASA Aeronautics Research Mission Directorate's Transformative Aeronautics Concepts Program.

References

[1] Tian, Y., Liu, K., Ok, K., Tran, L., Allen, D., Roy, N., and How, J. P., "Search and Rescue under the Forest Canopy using Multiple UAS," *International Symposium on Experimental Robotics (ISER)*, 2018.

[2] "Five Innovators Share 2019 AUVSI XCELLENCE Humanitarian Award," `https://www.auvsi.org/five-innovators-share-2019-auvsi-xcellence-humanitarian-award`, 2019. Accessed: 2020-11-24.

[3] Allen, B., and Alexandrov, A., "Ownship Self-Assessment of Success for Autonomous Search and Rescue Operations," *AIAA Aviation*, 2019.

[4] Bullo, F., Cortés, J., and Martínez, S., "Distributed control of robotic networks: a mathematical approach to motion coordination algorithms," Tech. rep., PRINCETON UNIV NJ, 2008.

[5] Murray, R. M., "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 129, No. 5, 2007, pp. 571–583.

[6] Cao, Y., Yu, W., Ren, W., and Chen, G., "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, Vol. 9, No. 1, 2013, pp. 427–438.

[7] Tsitsiklis, J. N., "Problems in decentralized decision making and computation." Tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND DECISION SYSTEMS, 1984.

[8] Tsitsiklis, J., Bertsekas, D., and Athans, M., "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, Vol. 31, No. 9, 1986, pp. 803–812.

[9] Jadbabaie, A., Lin, J., and Morse, A. S., "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on automatic control*, Vol. 48, No. 6, 2003, pp. 988–1001.

[10] Fax, J. A., and Murray, R. M., "Information flow and cooperative control of vehicle formations," *IEEE transactions on automatic control*, Vol. 49, No. 9, 2004, pp. 1465–1476.

[11] Olfati-Saber, R., and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, Vol. 49, No. 9, 2004, pp. 1520–1533.

[12] Ren, W., and Beard, R. W., "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on automatic control*, Vol. 50, No. 5, 2005, pp. 655–661.

[13] Moreau, L., "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on automatic control*, Vol. 50, No. 2, 2005, pp. 169–182.

[14] Reynolds, C. W., "Flocks, Herds and Schools: A Distributed Behavioral Model," *ACM SIGGRAPH Computer Graphics*, Vol. 21, No. 4, 1987, pp. 25–34.

[15] Leonard, N. E., and Fiorelli, E., "Virtual leaders, artificial potentials and coordinated control of groups," *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, Vol. 3, IEEE, 2001, pp. 2968–2973.

[16] Ogren, P., Fiorelli, E., and Leonard, N. E., "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic control*, Vol. 49, No. 8, 2004, pp. 1292–1302.

[17] Weisbin, C., Matthies, L., Rodriguez, G., Blitch, J., Krotkov, E., Lavery, D., and Shoemaker, C., "Miniature robots for space and military missions," *AIAA Defense and Civil Space Programs Conference and Exhibit*, 1998, p. 5309.

[18] Rybski, P. E., Papanikolopoulos, N. P., Stoeter, S. A., Krantz, D. G., Yesin, K. B., Gini, M., Voyles, R., Hougen, D. F., Nelson, B., and Erickson, M. D., "Enlisting rangers and scouts for reconnaissance and surveillance," *IEEE Robotics & Automation Magazine*, Vol. 7, No. 4, 2000, pp. 14–24.

[19] Curtin, T. B., Bellingham, J. G., Catipovic, J., and Webb, D., "Autonomous oceanographic sampling networks," *Oceanography*, Vol. 6, No. 3, 1993, pp. 86–94.

[20] Marden, J. R., Arslan, G., and Shamma, J. S., "Cooperative control and potential games," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 39, No. 6, 2009, pp. 1393–1407.

[21] Marden, J. R., Young, H. P., Arslan, G., and Shamma, J. S., "Payoff-based dynamics for multiplayer weakly acyclic games," *SIAM Journal on Control and Optimization*, Vol. 48, No. 1, 2009, pp. 373–396.

[22] Chasparis, G. C., and Shamma, J. S., "Distributed dynamic reinforcement of efficient outcomes in multiagent coordination and network formation," *Dynamic Games and Applications*, Vol. 2, No. 1, 2012, pp. 18–50.

[23] Marden, J. R., and Shamma, J. S., "Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation," *Games and Economic Behavior*, Vol. 75, No. 2, 2012, pp. 788–808.

[24] Lim, Y., and Shamma, J. S., "Robustness of stochastic stability in game theoretic learning," *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 6145–6150.

[25] Luo, J., Cooper, J., Cao, C., and Pham, K., "Cooperative adaptive control of a two-agent system," *International Journal of Control*, Vol. 86, No. 1, 2013, pp. 127–138.

[26] Luo, J., "Adaptive Control for Distributed Multi-Agent Coordination," 2013.

[27] Luo, J., and Cao, C., "Flocking for multi-agent systems with unknown nonlinear time-varying uncertainties under a fixed undirected graph," *International Journal of Control*, Vol. 88, No. 5, 2015, pp. 1051–1062.

[28] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del R'ıo, J. F., Wiebe, M., Peterson, P., G'erard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E., "Array programming with NumPy," *Nature*, Vol. 585, No. 7825, 2020, pp. 357–362. doi:10.1038/s41586-020-2649-2, URL `https://doi.org/10.1038/s41586-020-2649-2`.

[29] Cooper, J. R., "Optimal Multi-Agent Search and Rescue Using Potential Field Theory," *AIAA Scitech 2020 Forum*, 2020, p. 0879.

[30] Robbins, H., and Monro, S., "A stochastic approximation method," *The annals of mathematical statistics*, 1951, pp. 400–407.

[31] Puig-Navarro, J., Hovakimyan, N., and Allen, B. D., "Time-coordination strategies and control laws for multi-agent unmanned systems," *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 3990.

[32] Puig-Navarro, J., Hovakimyan, N., Alexandrov, N., and Allen, B. D., "Silhouette-informed trajectory generation through a wire maze for UAS," *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3845.