2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Performance Characterization of an Algorithm to Estimate the Search Skill of a Human or Robot Agent*

Audrey Balaska and Jason H. Rife, *Member, IEEE*
School of Engineering
Tufts University
Medford, MA 02155 USA
Email: Audrey.Balaska@tufts.edu

Abstract— **This paper characterizes an algorithm that estimates searcher skill level to support planning for search activities involving heterogeneous robot and human/robot teams. Specifically, we use Monte-Carlo simulations to determine the empirical accuracy of the estimator, to assess the quality of its predicted distribution (nonparametric) of agent skill levels, and the convergence rate of the estimate. The simulation study suggests that a single challenging search task can be used to estimate searcher skill within about 10%; however, the quality of the estimate is higher when searcher skill is high.**

I. Introduction

Search is a common goal in many robot applications. For example, humans are increasingly coordinating their efforts with robots to enhance search activities ranging from search and rescue missions [1]-[2], where the objective of the search is to find people who are lost or injured, to marine data collection [3], where the goal is often to locate and explore key environmental features. Increasingly, robots will also operate autonomously. This is the goal, for example, of an emerging class of consumer robots, including those that vacuum dirt or remove weeds [4]-[5].

Extensive prior research has been conducted to model and demonstrate search behaviors for robot and mixed human/robot teams. Many challenges exist, including reliable communication within the team [6]-[7], shared understanding of the terrain [8], and physical implementation of hardware that enables object detection and retrieval in natural terrain [9]-[10]. Another significant open challenge is to assimilate the data generated across human-robot teams over time [11]-[12]. This process is critical to enable searchers to adjust their motion planning and to enable supervisors to make decisions about how long to continue a difficult search. The process of fusing search data across the human-robot team is the primary focus of this paper.

In particular, we focus in this paper on search applications involving discovery of point objects. Detecting point objects requires direct observation, in contrast with estimating a spatially varying scalar field (e.g. temperature), where "hotspots" can be estimated by indirect observations in the general vicinity [13]-[17]. Even when a direct line of sight exists, there is no guarantee that the searcher detects a point object. As such, some researchers have modeled the detection of point objects as a probabilistic process, where missed-detection and/or false alarm probabilities are assigned to model detections errors (e.g. due to sensor noise) [11]. In our own recent work, we have adapted these probabilistic detection models to introduce missed-detection probabilities that are dependent on external factors, such as the skill of the searchers (in multi-agent human/robot teams) or the complexity of the environment [12]. Our modeling approach is compelling because it offers the flexibility to estimate the unknown external conditions (e.g. searcher skill, terrain probability) that influence search success without relying on additional information about the searcher. The resulting state distributions are strongly non-Gaussian, however, so additional work is needed to assess algorithm performance.

The main goal of this paper is to analyze the performance of the algorithm in order to evaluate its performance in assessing the skill of a human or robot agent. In doing this, there is then information to better interpret the estimator outputs. Our approach will be to use simulations to characterize estimator performance. To this end, we are less interested in characterizing mission outcomes (e.g. success in rapidly evacuating victims [18]) than in characterizing the accuracy of estimated parameters describing environmental conditions like the number of objects hidden in the search domain or the relative skill of heterogeneous searchers (e.g. robot and human). In addition to our analysis of the algorithm's accuracy, we also consider the quality of its estimated probability distribution and its convergence over sequential data updates.

The rest of the paper is organized as follows. First there will be a detailed summary of the model and estimation method we developed previously in [12]. Next, we will explain the metrics and procedures used for performance characterization. After a short description of Monte Carlo testing approaches, we present and discuss simulation results. The final section of the paper will discuss our conclusions from this work.

II. Model and Estimation Algorithm

This section describes our search model and an associated algorithm to estimate parameters relevant to the search, as first presented in [12]. Our search model is tuned to domains in which complete search coverage is not possible, meaning that the search domain cannot be exhaustively inspected in an allotted time. Given that complete coverage is not possible, search must be considered probabilistically, to capture the chance that hidden objects are detected even when an exhaustive search is not possible. To assign detection probability, we consider three elements of the task: the hidden objects themselves, the searcher, and the terrain. In general, these parameters may not be known a priori, so the

*Research supported by Air Force grant AFOSR FA9550-18-1-0465.

purpose of the estimator is to infer their values. The estimator might, for instance, infer the skill level of individual searchers (in order to optimize their deployment) or, alternatively, infer the number of objects that have yet to be found (in order to determine the marginal value of continuing the search).

In our model, we quantify detection probability for each object as a function of a terrain complexity parameter T and a searcher skill level S, where searchers can be human or robot agents. In covering a large area, a searcher must choose to focus more attention on certain spots than others. Searcher skill influences this scoping process, in which the searcher intentionally selects a subset of possible hiding spots to check. To this end, we quantify searcher skill as a probability S that the search is well scoped. (For instance, at present, an experienced human searcher is likely to be more perceptive in scoping a search than a capable robot, such that S would be higher for the human than the robot.) Even if the search is well scoped, some luck is required, as the terrain may obscure the object from the searcher's point of view. To this end we quantify the terrain complexity T as a conditional probability of missed detection given that the search was correctly scoped. Thus, as summarized in Fig. 1 (from [12]), a successful detection requires both that the searcher correctly scope which spots to check and, in those spots, interpret how the terrain might obscure an object of interest.

Implicit in our detection model is the notion that the searcher must be near the search object in order to find it. To assess proximity, we discretize the full search domain into a set of grid cells. We assume that an object can only be found by a searcher in the same grid cell g, and that each cell has a distinct terrain complexity $T(g)$. Though it is not a necessary restriction, we assume in this paper that the search objects are scattered sufficiently that at most one is present in any grid cell and that detection probabilities are independent for each object; for further simplicity, we model only the probability of missed detection and not also the probability of false alarms.

An example search area of a garden is represented in Fig. 2. Here, it is seen that the grid cells can be varying shapes and sizes, which reduces the restrictions on the possible domains that can be searched. In this paper, we do not consider methods for how to decompose the terrain into a set of grid cells; rather, we simply assume that a method exists

	Correct Scope	Incorrect Scope
Detection	$(1 - T(g)) \cdot S(i)$	0
Missed Detection	$T(g) \cdot S(i)$	$1 - S(i)$

Figure 1. Successful detection depends on the probability S that the searcher successfully scopes which spots to search and on the probability T that complex terrain obscures the object of interest, referred to in this paper as terrain complexity. Each robot or human searcher i may have a different skill level $S(i)$. The search domain is divided into grid cells g, each with its own terrain complexity $T(g)$.

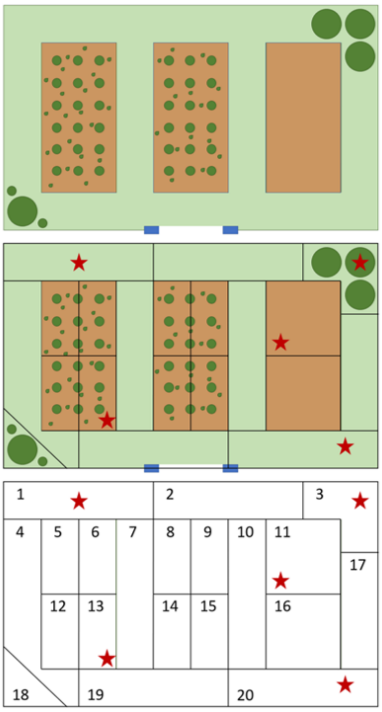

Figure 2. Image representation of discretizing a garden into grid cells. Object locations are marked using stars.

that can grid the search domain such that searchers will spend roughly the same duration of time Δ to search each cell and such that the terrain complexity is roughly uniform within each cell. In this paper, we assume that objects remain stationary (e.g. at locations marked by a star in Fig. 2).

Using this model, an estimation method was developed in [12] to predict unknown model parameters given the search history. The search history y_k is the record of all detections and non-detections in each grid cell through the current time k. In general, the estimator can be configured to process y_k in order to infer a wide range of parameters, including the total number of objects N (including those still hidden), the likely configurations C of hidden objects, the skill level S of each searcher, and/or the terrain complexity T of each grid cell.

For the case of estimating N, S, or both, [12] derives an appropriate form of Bayes rule as:

$$P(C,S|y_{k+1},T) = \frac{P(Y_{k+1}|C,S,y_k,T) \cdot P(C,S|y_k,T)}{P(Y_{k+1}|y_k,T)} \qquad (1)$$

Here the parameters include skill level S (a vector of values describing the skill levels of each agent), the terrain complexity T (a vector of values describing the complexity of each grid cell), and the configuration vector C (which is a hypothesis of the number of objects in each grid cell – either 0 or 1 in this paper). The left side of the equation is the posterior joint probability distribution function (PDF) of the configuration C and the skill S, given the latest set of observations Y_{k+1} from time $k+1$. The posterior is computed by multiplying two distributions, the prior $P(C,S|y_k,T)$ and the measurement probability $P(Y_{k+1}|C,S,y_k,T)$, where the measurement probabilities are defined by Fig. 1 for cells

containing objects (as specified by C) and zero otherwise. The denominator of the right-hand side is a normalization constant that ensures the integrated probability is one.

An important detail is that, even though the detection probabilities are independent for each object, the parameters determining the detection probability are related across cells (e.g. detection probability depends on S, which is a function of the searcher not the cell); as such, Bayes rule must consider the joint locations of objects across the search domain as described by C. Because the space of configurations is a powerset (2^G combinations for a set of G grid cells), it is useful to visualize the estimator results in a lower dimensional space by only considering the estimated total number of objects N in the domain (both found and unfound). The number of objects in the domain can be obtained for visualization purposes as the distribution:

$$P(N, S|_{y_{k+1}}, T) = \sum_{\{C \mid \|C\|_1 = N\}} P(C, S|_{y_{k+1}}, T) \tag{2}$$

The performance of our algorithm has not previously been characterized. In this paper, we will use simulation to characterize extensively the algorithm's performance on a simple model problem, that of a single agent searching when the number of hidden objects is known. This emphasis on the simple case is useful as a guidepost to understand how the algorithm performs as the parameter estimation grows in complexity. Importantly, keeping our performance assessment focused on the estimation of a single parameter, the searcher skill level S, allows for a deeper investigation and quantification than would be possible in an implementation estimating a large number of parameters.

III. Defining Performance Characteristics

This section defines "good" performance for the Bayesian estimator defined by (1). We are particularly interested in the skill-level PDF generated by the algorithm, so ideally the estimate of skill level S is correct. The broader predicted distribution of S should also be correct; moreover, the algorithm should converge quickly. The rest of this section describes tools for quantifying these characteristics: accuracy, distribution quality, and convergence.

A. Empirical Accuracy

Since the main purpose of our method is to compute a predicted probability distribution function (PDF) for the parameter or parameters of interest, it is important to evaluate whether the resulting PDFs are in fact centered on the simulated ground truth. In our analysis, we will compute two metrics representing the distribution average, and we will compare those metrics to a ground truth using simulation. The two metrics are the *mean* and *mode* of the parameter PDF.

In implementing the estimator (1), we use a histogram approximation, where each state is discretized with a finite number of values. The mean can be computed by integrating the probability weighted by those values. For instance, the predicted mean of skill is $\bar{S}$:

$$\bar{S} = \sum_q S_q \cdot P(S_q) \tag{3}$$

Here the overbar indicates the mean operator. The mean sums over all discrete values q of the skill level parameter S, where the size of q is determined by the discretization of possible skill levels. By contrast, the mode S^* of the same distribution is the skill level with the highest probability:

$$S^* = \text{argmax}_q\big(P(S_q)\big) \tag{4}$$

For each of the two metrics we can define an associate error by differencing the metric from ground truth. We quantify accuracy by taking the standard deviation of the error.

B. Distribution Quality

Whereas accuracy compares the best estimated value to ground truth, another approach is needed to assess how well the algorithm assesses its confidence in this estimate. To this end, a standard approach might be to compare the predicted standard deviation to the empirical standard deviation (as computed in the accuracy assessment above); such an approach makes sense for a linear estimator like a Kalman Filter. For our algorithm, however, the measurement update is strongly nonlinear, so the predicted PDF is dependent on the measurements. In other words, if the algorithm is run ten times with independent measurement noise in each case, then the algorithm will predict ten different PDFs for the parameter of interest.

In order to compensate for the variability of the estimated PDF, we map the ground truth backward through the predicted PDF, a process which ideally results in a uniform distribution of mapped errors. Consequently, by assessing how closely the mapped error distribution approximates a uniform distribution, we can assess the quality of the predicted PDF. The higher the quality of the PDF, the more confidence we have that the algorithm performs well.

C. Convergence

Because the estimator is recursive, there is an inherent transient, during which time the estimator converges toward a stable estimate over subsequent iterations k as new data becomes available. In order to determine how quickly the estimator converges, we assess the number of iterations required to arrive at a consistent distribution (meaning the distribution is no longer appreciably changing).

IV. Performance-Evaluation Simulation

To evaluate the performance characteristics defined above, we used Monte Carlo (MC) simulations. This simulation-based approach allows us to have full insight into algorithm performance, as the ground truth is known precisely and unambiguously, and as an arbitrary amount of statistical data can be generated simply by running the algorithm through a sufficient number of MC trials (or *runs*).

In this paper, the empirical accuracy and distribution quality are evaluated using 500 MC trials.

The baseline simulation consists of a search domain of 16 grid cells, 10 hidden objects, and one agent (robot or human), which moves through each grid cell in sequence. Although our approach can model multiple heterogeneous agents, we consider only a single agent in order to simplify interpretation of the simulation results. The agent's trajectory loops through all grid cells over and over again (conceptually switching direction each time). Each loop through all grid cells is labeled a pass. The PDF for the estimated parameter is updated at the end of each pass k, using (1). By default, each trial represents the agent as starting with a noninformative (uniform) prior then conducting 20 passes through the search domain. This number of passes was selected as low enough to manage computational load while still allowing most (if not all) of the ten hidden objects to be found.

The baseline configuration uses a terrain complexity of $T = 0.2$ and a searcher skill level of $S = 0.8$. These values were perturbed to promote analysis of the accuracy and distribution quality. In the case of our accuracy analysis, we elected to leave searcher skill at the baseline value but vary the terrain complexity through three cases {a,b,c}, each with T set according to Table I. In the distribution quality analysis, we introduced three more cases {d,e,f}, all with terrain complexity set to the baseline value but with skill level S set according to Table II. In the convergence analysis, we returned to case {c} but increased the number of passes from 20 to 500, ensuring that all objects were found.

For all configurations, the relevant output of each MC trial was a distribution over the skill level: *P(S)*. It is worth noting that in [12] this distribution is in fact computed from an estimated joint PDF over two parameters: *P(N,S)*. The joint marginal is computed by iterating equation (1) over all passes and then applying (2) after the final pass. The joint distribution was gridded with 21 uniformly spread values of S over the range 0 to 1 and 17 uniformly spread values of N over the range 0 to 16. By choice, we modified the estimator and provided it with information about the true value of C with a total object count equal to the true value of N. This choice intentionally reduced the dimensions of the problem, allowing us to gain more insight in the performance analysis by focusing on estimating a single parameter, the skill level S.

V. Results and Discussion

This section presents the results of our performance evaluation study. To provide context for these performance

TABLE I. Simulation Parameters for Accuracy Analsyis

Case:	a	b	c
Terrain Difficulty *T*	0.2	Uniformly random from 0-1	0.8

TABLE II. Simulation Parameters for Distribution Quality Analsyis

Case:	d	e	f
Skill Level *S*	0.2	0.5	0.8

analyses, which compile data across an ensemble of MC trials, it is perhaps first useful to visualize the outputs of individual trials. To this end, consider that the primary output of the estimation algorithm is a PDF of skill level S. A different PDF is generated by each trial. Examples of these PDFs are shown in Fig. 3 for simulation case e.

The figure clearly shows that *P(S)* varies from one trial to another. What is perhaps curious is that there are only a limited number of *P(S)* curves, less than 50 curves for 500 MC trials. One might expect that each MC trial would result in a unique *P(S)* distribution, but many clearly overlapped, which resulted in a smaller number of distinct distributions. Our hypothesized explanation for the overlap relates to the fact that the terrain complexities were all assumed equal for case e. With equal terrain complexities, the S distribution is influenced only by the total number of detections per time step and not the location of those detections; hence, we see each curve corresponds to a particular combination of detections per time step, regardless of the locations of those detections. It is worth noting that the multiplicity was higher for curves in the middle of the figure (with their peaks near the correct estimate).

A. Empirical Accuracy

To assess accuracy, we ran 500 MC trials each for cases {a,b,c}. For each trial, we computed best estimates of S in two ways – both as a mean and a mode of the estimated marginal *P(S)*. For the three cases of interest, Table III summarizes statistics of the skill estimation error, defined as best estimate minus truth.

The first column of the table provides the range of the error values, with the data more tightly spread than the theoretical limits of $[-S_{true},\ 1-S_{true}]$. The second column of data describes the average error, which is on the order of the grid discretization ($\Delta S = 0.05$) for the mean-based errors and smaller for the mode-based errors; these results do not establish that the estimator is unbiased, but they do suggest that estimation bias, if it exists, is small.

Lastly, a third data column computes the standard

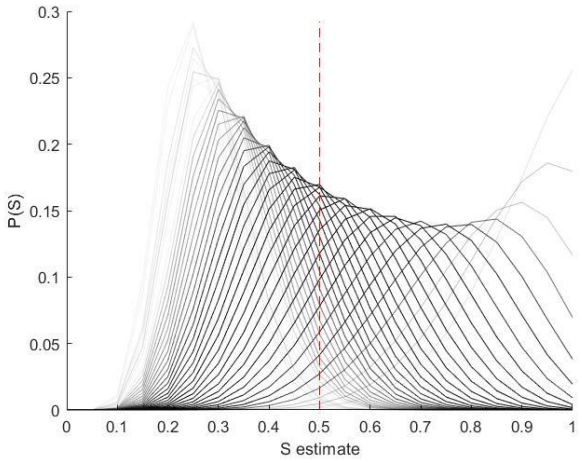

Figure 3. Estimated distribution of skill level for case e. S stands for skill level and *P(S)* stands for the distribution over the skill level. Each curve represents the estimated P(S) computed from one of 500 MC trials. The red dashed line represents the ground truth, which is S = 0.5 for case e.

TABLE III. EMPIRICAL ACCURACY SIMULATION RESULTS

	Error Range	Average Error	σ_d of Error
Case a Mean Errors	[-0.138, 0.370]	0.0296	0.0979
Case a Mode Errors	[-0.2, 0.4]	-0.0187	0.131
Case b Mean Errors	[-0.119, 0.436]	0.0474	0.103
Case b Mode Errors	[-0.2, 0.45]	-0.0109	0.161
Case c Mean Errors	[-0.117, 0.408]	0.0561	0.103
Case c Mode Errors	[-0.2, 0.45]	-0.0012	0.171

deviation of the errors. The standard deviation, labeled σ_d, is on the order of $\sigma_d \sim 0.1$ for the mean-based error and $\sigma_d \sim 0.15$ for the mode-based errors. This indicates that the skill estimates were accurate within 10-15% of the full range of possible values. The fact that the mode-based errors are more uncertain than the mean-based errors suggests that it is worth using the mean to derive a best estimate from *P(S)*, rather than simply picking the peak value (or *mode*) of the PDF.

A final observation is that the accuracy evaluation does not seem to show any significant dependence on terrain complexity. Whether the terrain complexity is low (case a), random (case b), or high (case c), the error statistics are very similar in all cases.

B. Distribution Quality

Our next question involves whether the predicted distribution *P(S)* is a good indicator of the actual errors. Because the predicted distribution changes significantly from one MC trial to the next, as shown in Fig. 3, we mapped the ground truth through the computed distribution for each trial (or more specifically through its inverse cumulative distribution function). This operation generates mapped errors that should ideally be uniformly distributed, assuming that the estimated PDF is correct. By contrast, if the estimated PDF is too wide, the empirical CDF will be too concentrated (with a central slope higher than that of a uniform distribution); if the estimated PDF is too narrow, the empirical CDF will be too diffuse (with a central slope shallower than that of a uniform distribution).

Results of this procedure are shown in Fig. 4 for cases {d,e,f} for 500 trials. In the figure, the mapped errors are sorted and assigned uniform probabilities, then integrated to construct an empirical CDF (stepped curve).

As shown in the figure, the correctness of the estimated PDF is strongly dependent on the skill level *S*. Of the three cases considered, case f (with $S = 0.8$) resulted in the highest distribution quality, with the mapped empirical CDF lining up very well with the reference line (right side of Fig. 4). As skill level decreases to $S = 0.5$ (figure middle) or $S = 0.2$ (figure left), the mapped error distribution becomes increasingly steep and biased toward the right. The steepness of the mapped distribution indicates that the estimated *P(S)* distribution is too wide. There is also some indication at lower skill levels of bias in the empirical CDF, and hence the *P(S)* estimate. The skill level also influences the step size of the curves show in Fig. 4. For example, in the case of a highly skilled searcher, all objects were found quickly; this meant that there was less variation in the PDFs compared to the MC simulation results for a poorly skilled searcher, where the detection of objects was less of a certainty.

The conclusion from this distribution-quality analysis is that the estimator appears to perform well, when skill level is high; however, its performance degrades continually as searcher skill level becomes lower.

C. Convergence

A final question involves the rate at which the simulation arrives to its final estimate. In our baseline simulation cases, we assumed the searching agent made 20 passes through the search domain before stopping; for the convergence study, we allowed the simulation to continue for 500 passes, which ensured convergence in the sense that no value of *P(S)* changed by more than 0.1% between sequential passes.

With the added duration of each trial, we ran only a limited number of trials to study convergence. For instance, Fig. 5 shows a set of 6 extended runs for simulation case c, which was a case with particularly challenging terrain ($T = 0.8$ for all grid cells). The figure shows the best mean-based estimate of skill *S* on the vertical axis as a function of the number of completed passes *k* on the horizontal. As shown in the figure, most trials converged well (i.e. with 5% of the final answer) by the completion of 10 to 20 passes; however, one of the trials shown required nearly 40 passes to converge well.

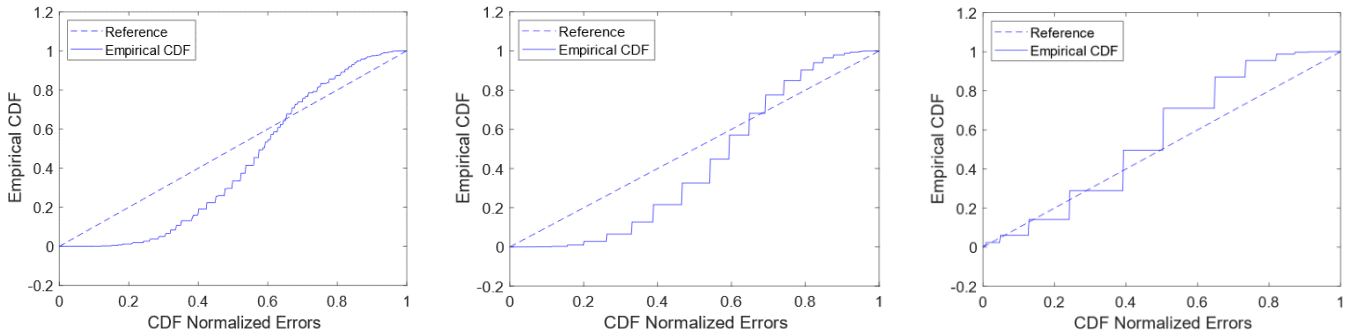

Figure 4. Normalized Cumulative Distribution Function (CDF) results for varying skill levels: (left) case d, with *S*=0.2, (middle) case e with *S*=0.5, (right) case f with *S*=0.8.

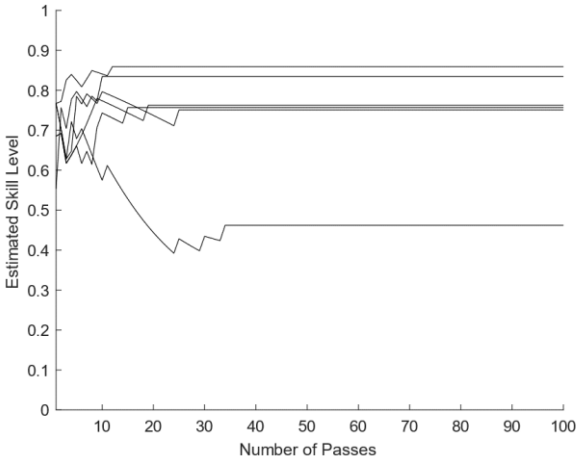

Figure 5. Converge for six trials of case c.

Based on Fig. 5 and other cases we have run, we observe empirically that convergence time increases with more challenging terrain (high *T*) and with less searcher skill (low *S*). A more systematic study will be required to precisely quantify the relationship between convergence time, *T*, and *S*. One observation of interest, however, is that the convergence rate provides a strong clue about the utility of acquiring further information to enhance the estimate.

VI. Future Work

This paper provides an indication that our estimation method can be used to evaluate the search skill of humans and robots in order to determine how they might best be deployed in a heterogeneous human/robot search team. Nonetheless, significant opportunities exist to continue to improve the methodology. Questions that might lead to expanded utility include the following:

- Do the results of the single-agent study presented here extend to the multi-agent case?
- How well can the algorithm estimate the number (and potential location) of remaining objects that have not yet been found?
- What are the limits of observability in the model? (For example, what combinations of the parameters *T*, *S*, and *N* can be estimated simultaneously?)

As the model is better understood, it is expected that the estimator will be tested in an experimental setup.

VII. Conclusions

In this paper, we analyzed an algorithm to estimate searcher skill for individual agents in a human/robot team. By characterizing a one-agent simulation, we found that the estimator could determine the skill level of the searcher within a standard deviation of approximately 10% of full range. The estimator produced a distribution of possible skill levels, and this distribution was representative of error data generated via Monte Carlo simulations as long as the searcher's skill level was relatively high. When the searcher's skill level is low, however, the predicted skill-level distribution becomes somewhat less predictive of the actual error distribution (wider and with a small bias). The convergence rate of the skill estimate depends strongly on the amount of data available, but many repeated search passes may be required before final convergence of the estimate.

References

[1] I. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, "Human-Robot Teaming for Search and Rescue," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 72–78, 2005.

[2] M. A. Goodrich, J. L. Cooper, J. A. Adams, C. Humphrey, R. Zeeman, and B. G. Buss, "Using a Mini-UAV to Support Wilderness Search and Rescue: Practices for Human-Robot Teaming," *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, 2007.

[3] T. Somers and G. A. Hollinger, "Human–robot planning and learning for marine data collection," *Autonomous Robots*, vol. 40, no. 7, pp. 1123–1137, Mar. 2015.

[4] J. L. Jones, "Robots at the tipping point: the road to iRobot Roomba," *IEEE Robotics & Automation Mag.*, vol. 13, no. 1, pp. 76–78, 2006.

[5] T. Gross, "Garden-Variety Robot," ASEE Prism, vol. 26, no. 4, p. 12, Dec. 2016.

[6] R. E. Giachetti, V. Marcelli, J. Cifuentes, and J. A. Rojas, "An agent-based simulation model of human-robot team performance in military environments," *Systems Engineering*, vol. 16, no. 1, pp. 15–28, 2012.

[7] Murphy, Robin R., Kevin S. Pratt, and Jennifer L. Burke. "Crew roles and operational protocols for rotary-wing micro-UAVs in close urban environments." *ACM/IEEE Conf. on Human Robot Interaction*. 2008.

[8] T. Kaupp, B. Douillard, F. Ramos, A. Makarenko, and B. Upcroft, "Shared environment representation for a human-robot team performing information fusion," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 911–942, 2007.

[9] Niedzielski, T., Jurecka, M., Miziński, B., Remisz, J., Ślopek, J., Spallek, W., Witek-Kasprzak, M., Kasprzak, Ł. and Świerczyńska-Chlaściak. "A real-time field experiment on search and rescue operations assisted by unmanned aerial vehicles." *Journal of Field Robotics*, vol. 35, no. 6, pp.906-920, 2018.

[10] Gu, Yu, et al. "Robot foraging: Autonomous sample return in a large outdoor environment." *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 93-101, 2018.

[11] J. Ma and J. W. Burdick, "A probabilistic framework for stereo-vision based 3D object search with 6D pose estimation," 2010 IEEE International Conference on Robotics and Automation, 2010.

[12] A. Balaska and J.H. Rife. "Development of a model and estimation method to represent team search with uncertain detection," *Proc. ION International Technical Meeting (ION ITM 2020)*, 2020.

[13] D. O. Popa, A. C. Sanderson, R. J. Komerska, S. S. Mupparapu, D. R. Blidberg, and S. G. Chappel. "Adaptive sampling algorithms for multiple autonomous underwater vehicles." *IEEE/OES Autonomous Underwater Vehicles*, pp. 108–118, 2004.

[14] E. Fiorelli, N. E. Leonard, P. Bhatta, *et al*. "Multi-AUV control and adaptive sampling in Monterey Bay." *IEEE J. Oceanic Engineering*, vol. 31, no. 4, pp. 935–948, 2006.

[15] N. Cao, K. H. Low, and J. M. Dolan. "Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms." *Autonomous Agents and Multi-agent Systems*, pp. 7–14, 2013.

[16] Y. Xu, J. Choi, S. Dass, and T. Maiti. *Bayesian Prediction and Adaptive Sampling Algorithms for Mobile Sensor Networks*. Springer, 2015.

[17] L. V. Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake. "Information-driven adaptive sampling strategy for mobile robotic wireless sensor network." *IEEE Trans. Control Systems Technology*, vol. 24, no. 1, pp. 372–379, 2016.

[18] A. Jacoff, E. Messina, B. Weiss, S. Tadokoro, and Y. Nakagawa, "Test arenas and performance metrics for urban search and rescue robots," Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453).