A Novel User-Guided Interface for Robot Search

Shahar Kosti[1], David Sarne[1] and Gal A. Kaminka[1,2]
[1]Computer Science Dept. and [2]Gonda Brain Research Center
Bar Ilan University, Israel

Abstract— **Human operators play a key role in robotic exploration and search missions, as the interpretation of camera images typically requires the visual perception skills of humans. Thus one the key challenges in building effective robotic systems for such missions lies in developing good operator interfaces. In this paper, we present a novel asynchronous user-guided interface for human operators of robotic search of an unknown area. Enabled by efficient methods to store and retrieve recorded images (and meta information) in real-time, our interface allows the operator to click on any point of interest. The operator is then presented with highly-relevant images that cover the point, without occlusion. This, in contrast with system-guided approaches, where an automated system selects areas and images for inspection. Experiments with 32 human subjects in two different-size maps favor the user-guided approach we present over the system-guided approach. Additional experiments with human subjects provide an explanation as to the environment characteristics that favor our approach.**

I. Introduction

There is a growing interest in human operator interfaces for robotic exploration and search in unknown areas [6], [9], [7]. Human operators play a key role in such missions, as the interpretation of camera images typically requires the visual perception skills of humans [9]. Thus a key challenge lies in building effective operator interfaces.

The traditional approach to operator interfaces for robot search is called *synchronous*, often relying on *camera guided teleoperation* [6]. Here, each robot carries a video camera, and a remote operator views live video from the robot (and if teleoperating it, also controls it, e.g., using a joystick). The operator is responsible for interpreting the images as they are received. Both tasks (navigation and interpretation) require constant attention and contribute to the operator's workload. Automating the navigation tasks within the mission reduces operator workload [8]. Once the robots are automatically controlled, it is possible to further reduce operator load by eliminating the need to view *live* video.

Asynchronous viewing interfaces rely on the autonomous operation of the robots, presenting the operator with recorded imagery rather than live (*synchronous*) video streams [6]. This has several advantages. First, it allows scaling up the number of robots with only a moderate increase to the operator's workload, since the operator no longer has to observe live video feeds from multiple robots [7]. It also overcomes communication constraints that often limit the ability to stream live video from the robots. With asynchronous interfaces, the transmission of recorded images is flexible, as images may be (re)transmitted with a delay. Images may even be transmitted in batches (e.g. whenever wireless reception allows).

This research was supported in part by ISF grant #1511/12.

A key question in asynchronous interfaces for robot-based search is how to select the most relevant imagery to display to the operator. A leading approach automates this process, and automatically selects images for viewing, based on a utility value, which is computed by the image area that was not already seen [7]. Users can then view images according to that order, and navigate through images that were recorded near the selected image. This is a system-guided approach, as the image sequence to view is determined by the system.

In contrast, we propose an asynchronous *user-guided* interface, aiming to speed up and improve the efficiency of asynchronous exploration of indoor environments based on recorded imagery. Our interface is based on having the operator select *points of interest* (POI) on a map and then having the interface provide her with highly-relevant images of that POI from several view points, after applying a dynamic filtering and ranking process over the recorded images. This way, the system utilizes the human's expertise and spatial capabilities for better reasoning about where to explore next, and the interface's computational capabilities for better reasoning about what images best cover the specific area of interest. To enable selection of images based on points that appear in them, we develop efficient methods for storage and retrieval of images, indexed by areas covered.

We evaluated our interface in several separate experiments. The main experiments contrasted user-guided and system-guided interfaces in a standard simulated USAR mission, used in other reported research [10]. 32 human subjects worked in two different indoor search environments. The results demonstrated that subjects using the user-guided interface performed better in terms of the number of found targets, in larger environments.

II. Related Work

Much of previous research on robotic search interfaces focused on *synchronous* interfaces. Here the operator has to constantly monitor one or more live video feeds, while performing mission-related tasks at the same time. Humphrey et al. [4] presented a synchronous interface design for multi-robot bomb defusing task. They showed that increasing the number of robots increased situational awareness, but also increased operator workload significantly.

To address the increase in operator workload, [8] introduced the concept of a *fulltask*, which is composed of the *exploration* (cover as much area as possible) and *perceptual search* (locate victims on video screens) sub-tasks. The results suggested that while the *exploration* sub-task is a good candidate for automation, it is possible to further reduce operator workload by eliminating the need for watching live videos, i.e., turning to *asynchronous* interfaces.

Two major concerns of asynchronous interfaces for multi-robot search, are how to select most relevant imagery of display, and how to display it to the operator. The most extensive work on asynchronous interfaces, explored several interfaces for search missions, where participants were required to mark disaster victims on a map. The work of Velagapudi et al. [6] compared *synchronous* and *asynchronous* interfaces. The asynchronous interface had no live video display. Instead, an operator directed the search task by assigning robots with map coordinates. After reaching the final coordinate, the robot would take a panoramic image of that location, which appeared as a new symbol on a map. The user would then click on the symbol to view a panoramic view of the recorded location. The experimental results showed that a better performance can be achieved with the *asynchronous* interface when the number of robots is relatively substantial.

More recently, [7] sought to insert automation in the perceptual search task. This work explored an *image queue* interface, where the images taken by all robots are stored in a database. At each time the images in the database are sorted according to the size of the area in the image that has not been presented to the operator in previously presented images. The results showed a decrease in workload and number of errors, but no improvement in number of found victims. In contrast, our approach (which uses the map as the primary interaction component, rather than an automated image queue) shows significant improvement in the number of victims found.

III. A User-Guided Asynchronous Interface

The Point Of Interest (POI) user-guided asynchronous interface has two main functionalities: view recorded imagery and mark locations on a map. In the context of a USAR mission, the marked locations can specify the approximate positions of disaster victims. The system architecture is based on two primary modules: *User Interface* and *Image Retrieval*. The first is responsible for getting user requests in the form of points of interest, and for displaying images. The latter is responsible for processing user input, searching through the recorded images and intelligently deciding on the images to display based on ranking. We begin by describing the *User Interface* module, and then move on to describe the *Image Retrieval* module.

The input to the interface is an image database collected by one or more autonomous robots, that provide an ongoing stream of camera images and associated range scanner readings. The latter can be used to build and maintain a map of the indoor environment utilizing a Simultaneous Localization and Mapping (SLAM) technique (e.g., [2]). In the displayed maps, white areas represent clear space, while black areas represent occupied space (walls, obstacles, etc.).

A. Navigating Between Images

The user navigates between recorded images using the user interface shown in Figure 1. The user selects a POI

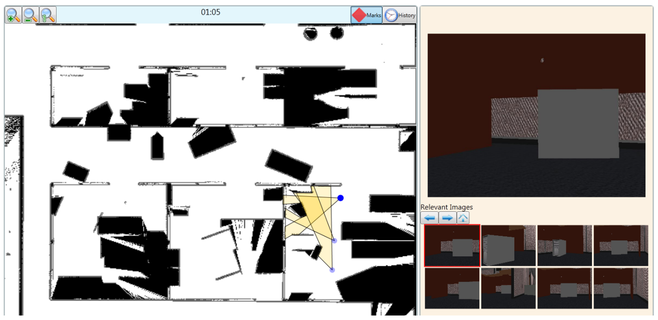

Fig. 1. The interface after a point of interest was selected.

on the map by clicking on it. The system finds all camera images that cover the selected point, and ranks them. The highest-ranked image is displayed in the upper-right corner. The bold dot on the map shows the robot location while recording the current image. This latter information is of much importance to the operator's orientation when scanning the image. Some other images are available for selection on the map, their locations displayed as smaller dots. All other images that cover the POI are displayed as thumbnails, below the current image. The user can select any of these to refine the automatic image selection that cover the POI.

We emphasize that the interface does not employ a specific map coverage strategy. It is up to the user to determine which areas to visit and in what way to do so. This way, we benefit from the user's inherent spatial and cognitive capabilities that often cannot be captured by a computerized system. For example, to distinguish between seen and unseen map areas, users are provided with a dedicated map layer that highlights those areas covered by previously presented images.

B. Storing and Finding Images

Recorded images from all robots are saved in a single database. Each database entry contains the image and auxiliary information regarding the robot at the time of recording. By storing the auxiliary information with each image, we save significant computation time when responding to queries. The following data is kept for each image:

- Location and orientation (heading) of the robot that generated the image, at the image recording time.
- Range scanner readings from the recording time, kept as a polygon that begins and ends at the robot's location. The polygon indicates which part of the map is covered by the recorded image, and allows computation of occlusion in the image.

Some adjustments have to be made to the polygon, so that it better represents the recorded image. A range scanner will typically have a wider field of view than a camera. To make sure we only use the map area that is covered by the image, the polygon is clipped to the actual camera field of view. Obviously, this is only an approximation of the visible map area, but in reality it provides adequate results.

During retrieval, as the user clicks on a point of interest p, the system must find all polygons that cover (contain) that point of interest. Testing whether a point is inside a polygon is a basic problem in computational geometry. We use the *crossing number* (CN) algorithm as described by O'Rourke

[5], also known as the *even-odd rule*. The challenge is that even this efficient algorithm must be called on all polygons. We discuss two approaches to solve this problem.

A simple improvement requires some form of preprocessing in the form of keeping for each polygon in the database also its bounding rectangle. Now, instead of running CN on all images, we run it only on polygons whose bounding rectangle contains p. This performs much better than the naïve approach.

A better technique is to use a spatial access method as the basis for data lookup. This reduces the number of polygons that are processed in response to each query. This approach is more efficient, and thus more appropriate as the number of images to be stored is larger. One example of a spatial data structure is R-Tree [3], a search tree that allows indexing of data by rectangles. When constructing the tree from polygons, we use their bounding rectangle as the key. Upon receiving a query for a point p, we first use the R-Tree to retrieve all polygons whose bounding rectangles contain p. Then, the CN algorithm is used in a similar manner to test whether p is actually contained in the polygon.

C. Ranking Images

Normally, the POI selected by the user would be covered by many images, possibly too many. For example, in the experiment reported in this paper each given point was covered by 10–300 images. Most of these images are redundant, as they cover overlapping areas. In order to deal with this extreme amount of information related to the point of interest we apply ranking. The ranking allows the operator to effectively observe the point of interest from different relevant perspective in a relatively short time, efficiently concluding whether a victim is present at the point of interest and if so where exactly. It is carried out as follows:

1) Find all images that cover the point of interest (p).
2) Group the images by robot heading and sector resolution r (see details below).
3) For each group, compute the utility value u of all images, and select the image with best u (see details below).

To provide the user with different perspectives of the POI, we group the images by their view angle. For each image we compute an angle θ, between the POI and the robot location while recording. The images are then grouped in sectors, considering the θ value and the resolution parameter r. This process is illustrated in Figure 2. The big dot is the point of interest, selected by the user. The small dots represent the images that cover the POI. Images are grouped in sectors by their θ angle, when the actual size of each sector is determined by r. We then pick one image from each group to the *best* group, and gather the rest in the *other* group.

The ranking process produces two sets of images: *best*, which contains the highest-ranked images of each sector, and *other*, which contains all other images that cover p. The highest-ranked image of *best* is displayed automatically, while the rest of the images are available for selection on the map. All images, including the *other* images, are displayed as thumbnails and ordered by their utility value.

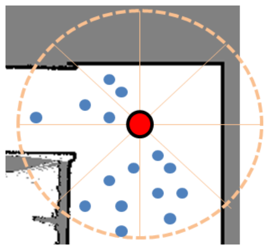
Fig. 2. Grouping images into sectors

D. Utility Value Computation

Determining whether an image provides a good view of the POI is not a trivial task, as many factors may be considered when doing so. Such factors include the area size covered by the image, distance from the POI, whether the image is directed towards the POI, whether the POI resides in a small room or in the middle of a corridor, and more. For this research, we have chosen to consider the following attributes when computing the utility value:

- Image area (area formed by the range scanner readings). The greater then better.
- Distance from the POI. The smaller the better.
- Image centering related to the POI. The image should be centered as much as possible. We define an image to be fully *centered*, if the robot faced directly towards the POI while recording. We this context, we consider the measure

$$centered[image] = |heading[image] - \theta|$$

which we attempt to minimize.

Finally, we use a weighted sum of the measures, according to the above attributes, as an indicator for the utility achieved from presenting this image:

$$u[image] = w_1 \frac{area\,[image]}{area_{max}} + w_2 \frac{distance_{min}}{distance\,[image]} + w_3 \frac{centered_{min}}{centered\,[image]}$$

where w_i is the weight of the i-th attribute.

Note that the minima and maxima are computed over each group (sector) separately. Note that this is only one example of a utility function for our method. Other functions, that consider different environmental parameters or are calculated differently, can be used without major changes to the other components of the system.

In the reported experiment, the parameters (weights) for the utility function were set in a pilot session. The procedure consisted of selecting a set of points in each of the maps used in the experiment, and comparing the coverage quality of the image ranking process with different weight values.

IV. Evaluation of the User-Guided Interface

In this section, we describe in detail the main experiment that was conducted in order to evaluate the user-guided, POI interface. We first describe the experimental design, along with a some details about the participants in the experiment. We then provide details as to the competing interface with which we contrast the POI interface, the experiment environments, and the performance measures.

A. Experimental Design

The experiment was intended to test the hypothesis that the POI interface facilitates improved search results, compared to the state-of-the-art system-guided interface (called *Best-First* hereinafter), presented in [7], and described below. The experiment design followed a between-subjects design, with each participant using only one of the interfaces. Participants performed the missions in a random order, to minimize learning effects.

32 adult students were recruited to this study, balanced for gender. Participants received a fixed show-up fee, along with a variable bonus based on the number of found victims. The average participant age was 23.2 ($SD = 3.8$). From a preliminary questionnaire aiming to check participants' computer skills, we found that our participants reported using a computer for an average of 6 ($SD = 3.2$) hours per day, and a smartphone device for an average of 4.4 ($SD = 4.7$) hours per day. The average reported daily duration for playing video games was 0.5 ($SD = 1.2$) hours. None of the participants had prior experience with robot systems.

Before participating, each subject read an instructions manual for the chosen interface, and performed a training session using a dedicated training map. Participants had to successfully mark three victims in the training session, in order to ensure they control the interface they are using and can perform the missions to follow. Afterwards, participants were given 10 minutes in each environment to locate and mark as many victims as possible. None of the participants were able to cover the entire area in this time.

B. The Competition: The Best-First Interface

The *Best-First* (originally called *image queue*) interface is presented in Wang et al. [7]. This is one of the leading interfaces for USAR tasks, a pinnacle of years of experiments by the authors with both synchronous and asynchronous interfaces of various designs. We have implemented this interface as described by the authors.

The key to the *Best-First* interface is that unlike the POI interface, the Best-First interface does not allow clicking on map points to see images. Instead, it is the system that selects images for viewing, based on its own calculations of the utility of different images. The user can use the "Next" and "Previous" buttons to go through a queue of images, ordered according to a utility measure computed by the size of area in the image that has not been covered in any of the images presented to the user so far. Table I summarizes the main differences between our interface and the *Best-First* interface.

	POI	*Best-First*
Image navigation	Select POI on map	Next/Prev button
Next area to cover	User decision	System decision
Thumbnails	Cover current image area	Recorded close to current image
Thumbnails order	Utility value	Recording time
History layer	Available	Not available

TABLE I
INTERFACE COMPARISON.

C. Simulated Environments

The reported experiment was conducted in simulated environments, generated by USARSim [1]. USARSim is a popular high-fidelity simulation package for USAR research, used as the basis for international USAR competitions, and human interface research [10]. It has been shown to accurately model robot behavior and sensors, in particular camera video and laser range scanner. This is the same simulation in which the *Best-First* interface was tested in [7].

Two simulated USAR environments ("maps") were created, based on an office environment from the 2006 RoboCup Rescue competition finals. 20 human-like characters ("victims") in various postures were manually placed in each environment, in rooms and corridors, using the USARSim environment editor. The environments differed in area size, the number of recorded images and the average image space covered by a victim ("visibility"). Visibility was computed by performing ray casting for each image, at a resolution of two degrees, and counting the number of hits for victims.

Map	Victims	Images	Area (m^2)	Visibility Mean (SD)
1	20	3209	327	5.01 (16.76)
2	20	4579	483	9.91 (30.45)

TABLE II
SUMMARY OF ENVIRONMENTS PARAMETERS

We used imagery and laser scan readings recorded by robots that explored the simulated environments. The resulting 2D maps generated by a SLAM process are presented in Figures 3 and 4. The same imagery and data was provided to the two interfaces. The differences between the two are summarized in Table II.

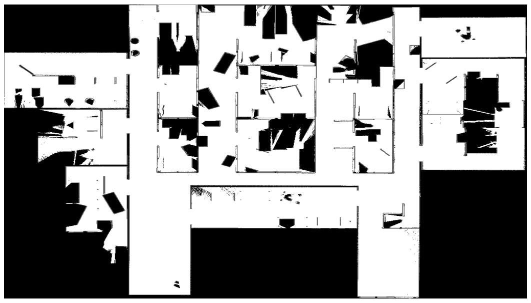
Fig. 3. Map 1, rotated 90°.

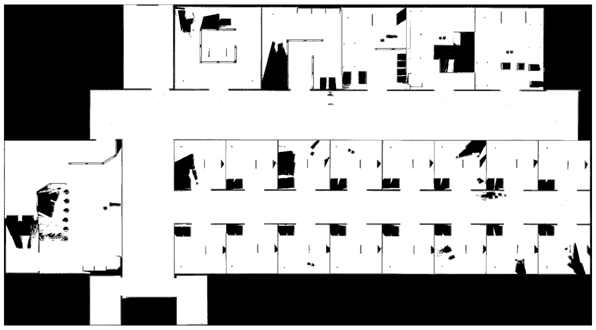
Fig. 4. Map 2.

D. Performance Measurement

There are many different variables that can be measured to assess the performance of subjects, along different lines.

Due to space limitations, we report on a subset of the results and thus only discuss a subset of the performance measures.

The main performance measure deals with subjects' ability to locate victims in the imagery, and correctly mark their location on the map. To do this, we needed a way to identify whether the locations marked by subjects matched the actual victim location in the map.

Prior to the experiment, a list of victim rectangles was generated for each environment. The rectangles indicated the map areas considered as victims, and were generated by matching victim locations in the USARSim environment editor with the 2D map. Each mark made by the participants to indicate a victim was assigned to the nearest bounding box. If a mark was within the bounding box or within $1m$ distance, it was considered successful for that victim. We tested various values for this distance threshold. The $1m$ accuracy choice was verified using a visual inspection, making sure that no anomalies occur with the experiment data when using this value. Such anomalies included marks that are considered successful, but seem too far from the victim (e.g., in another room), and marks that are not considered successful, but are close enough to be so.

Given the distance threshold—the rectangles surrounding each victim—we categorized each mark made by a subject as one of the following:

1) *found*: first successful mark for a victim.
2) *duplicate*: successful mark for a victim that was already successfully marked.
3) *false positive*: a mark that could not be assigned to any of the victims (based on the distance threshold).

In addition, we measured the number of *false negatives*, which considers victims that were seen by the participant sometime throughout the session, but were never marked. To compute this measure, we used the victim visibility data of the images that were seen by the participant. Note that since the sessions were time-limited, these measures along with the number of seen images, can be used to compare the rate at which participants performed the mission.

Several important measures can be computed by further processing of these basic measures. For instance, we look closely at the variance or standard deviation of the four basic measures above, and also at the rate at which they change with experiment time. We discuss these in the next section.

V. Results

We start by presenting the results using the main performance measures defined earlier such as found victims. We then present a follow-up experiment that was conducted in order to explain some of the results.

A. Main Results

The aggregated results for found victims, false negatives, false positives, and duplicates, for the two maps, are shown in Figures 5 and 6. The vertical axis in each of the subfigures measures the result. In the *Found* subfigure, a higher result is better. In the others, a lower result is better. In each of the 4 subfigures, the left box plot shows the *Best-First* results, and the right box plot shows the *POI* results.

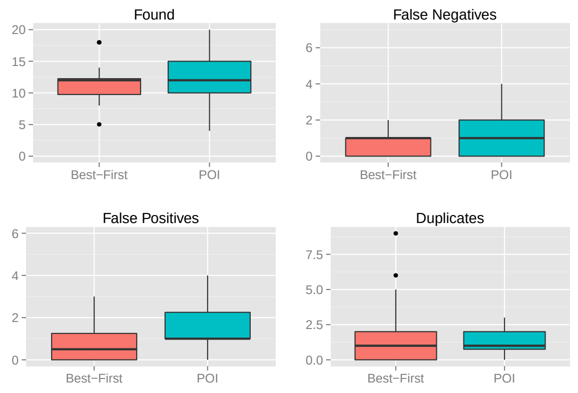

Fig. 5. Mark results—Environment 1 (Smaller, less visibility)

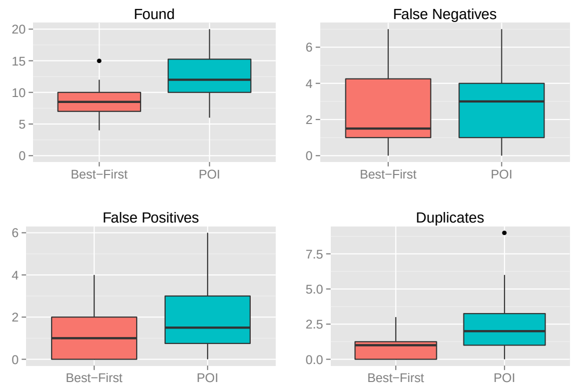

Fig. 6. Mark results—Environment 2 (Larger, more visibility)

The number of found victims: Overall, subjects who used the *POI* interface found (and correctly located) more victims than subjects using the *Best-First* interface. In environment 1, subjects using *POI* found 12.4 victims on average. Subjects using *Best-First* found 11.5. The difference is notable, but is not statistically significant (two-tailed t-test, $p = 0.1$). In environment 2, the number of *found* victims in the *POI* condition (12.7) was significantly higher than in *Best-First* condition (8.6) (two-tailed t-test, $p < 0.003$).

Measures of errors: The duplicates, and the false positives and negatives measure operator errors. In environment 1, the *POI* results are more varied than the *Best-First* results, but the were no statistically significant differences between the two conditions. In environment 2, The number of false positives and negatives did not differ significantly between the two conditions, however, the *POI* condition did have significantly more duplicates than *Best-First* (2.4 vs 0.9, respectively; $p < 0.028$)

Task progress and variance: Figure 7 shows task progress over time for both of the environments, by displaying the number of correct marks of all participants, individually, throughout the session. In environment 2, the number of correct marks for the *POI* condition increased in a higher rate than the *Best-First* condition, as evident by the regression line. Thus not only were more victims found using *POI*, but the rate of their detection was better.

However, as evident in the figures, the *POI* condition resulted in higher variance across the different measures, in both environments. On one hand, this indicates less consistency between operators, and thus greater need for training. On the other, it also shows more promise: The best-performing operators using *POI* were much better than the best-performing operators using *Best-First*.

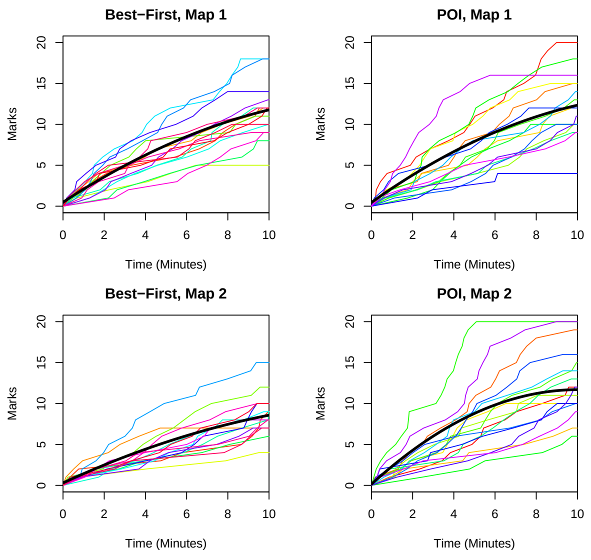

Fig. 7. Correct marks over session time. Each colored curve represents a different participant, while the black curve is a second order polynomial regression line. A "favorable" pattern in this chart is a curve that increases fast and reaches the maximal value.

B. POI *scales better in size*

A close look at the *found victims* results shows that the *POI* results were virtually identical in the two environments (two-tailed t-test, $p = 0.79$), while the *Best-First* results differed significantly ($p = 0.01$). At first glance this may seem perplexing: environment 2, where *Best-First*'s performance decreases, has higher victim visibility than environment 1 (recall Table II). We therefore hypothesized that rather than visibility, it was the size of the environment that had a detrimenal effect on subjects using *Best-First*, but not on subjects using *POI*. This hypothesis is not intuitive; the two interfaces differ in the images displayed to the subjects, and should therefore be sensitive to visibility. But there is no explicit way in which they respond differently to different-sized maps. Thus we did not expect the map size to be the factor causing the difference.

To test this hypothesis, we conducted a second experiment. Its goal was to demonstrate that visibility does not affect performance. In other words, that the human operators were able to quickly reach a correct decision on the presence of a victim in the image, essentially regardless of the size of the victim's body part within the image. If so, this would lend support to our hypothesis that it was the size of the map that affects *Best-First*, but not *POI*.

We recruited 16 participants from the same population. In both environments, we collected images from four visibility categories: *High* (a victim is clearly visible in image), *Medium*, *Low* (a fraction of body part in image) and *Empty* (no victim in image). Each of the participants was shown randomly-selected images from this collection, and had to decide for each image whether it contains a victim.

	Empty	Low	Medium	High
% errors	1 (2.2)	2 (4.5)	1 (2.8)	0 (0)
Time	1.3 (0.7)	1.2 (0.6)	1.2 (0.6)	1.1 (0.4)

TABLE III
VISIBILITY EXPERIMENT: MEAN (SD) OF ERRORS PERCENTAGE AND TIME (IN SECONDS) PER IMAGE.

Table III shows the mean number of errors and mean time spent on images, for each of the categories. In general, the mean error rate was low for all categories. Images from the *High* category required less time to handle, and the mean time difference from *Empty* images was 240 milliseconds.

These results suggests that visibility did not have a significant effect on mission performance (error rate and time spent viewing images) in the experiment described in Section IV. Since the other notable difference between the environments is their size, it lends support to our hypothesis.

VI. SUMMARY

We have presented a novel user interface for robotic search missions. The *POI* interface allows the user to determine which parts of the map to explore, without worrying about how images for this location are retrieved and sorted. We discuss image indexing and retrieval methods that allow this to take place in real-time, and ranking methods that find good images for the point of interest, maximizing coverage and minimizing occlusion.

We find several advantages in the *POI* interface. In comparison to the state of the art, a set of experiments with up to 32 human subjects demonstrated that it results in significantly improved search performance and success rate. It is also less sensitive to changes in map size, achieving consistent results regardless of map size. However, the results were highly variable over time, suggesting the need for improved training. We hope to address this in future work.

REFERENCES

[1] S. Carpin, T. Stoyanov, and Y. Nevatia. Quantitative assessments of USARSim accuracy. In *PerMIS'06*, 2006.
[2] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): part i the essential algorithms. *IEEE Robotics And Automation Magazine*, 2:2006, 2006.
[3] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *SIGMOD'84*, pages 47–57, 1984.
[4] C. M. Humphrey, C. Henk, G. Sewell, B. W. Williams, and J. A. Adams. Assessing the scalability of a multiple robot interface. In *HRI'07*, page 239, 2007.
[5] J. O'Rourke. *Computational Geometry in C (2nd Edition)*. Cambridge University Press, 1998.
[6] P. Velagapudi, J. Wang, H. Wang, P. Scerri, M. Lewis, and K. Sycara. Synchronous vs. asynchronous video in multi-robot search. In *ACHI'08*, pages 224–229, 2008.
[7] H. Wang, A. Kolling, N. Brooks, S. Owens, S. Abedin, P. Scerri, P.-j. Lee, S.-Y. Chien, M. Lewis, and K. Sycara. Scalable target detection for large robot teams. In *HRI'11*, pages 363–370, 2011.
[8] H. Wang, M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara. How search and its subtasks scale in n robots. In *HRI '09*, page 141, La Jolla, California, USA, 2009.
[9] H. Wang, P. Velagapudi, P. Scerri, K. Sycara, and M. Lewis. Using humans as sensors in robotic search. *FUSION'09*, pages 1249 – 1256, 2009.
[10] J. Wang, M. Lewis, S. Hughes, M. Koes, and S. Carpin. Validating usarsim for use in hri research. In *HFES'05*, pages 457–461, 2005.