Human-Robot Emergency Response - Experimental Platform and Preliminary Dataset

Technical Report #UM-CS-2014-006

Hee-Tae Jung, Takeshi Takahashi,and Rod Grupen
School of Computer Science,
University of Massachusetts Amherst,
Amherst, MA 01003-9264, USA
{hjung,ttakahashi,grupen}@cs.umass.edu*

July 28, 2014

Abstract

This paper presents progress towards a research infrastructure for studying human-robot performance in laboratory emergency response scenarios and a preliminary dataset. It incorporates an emergency response team that is composed of a human participant, $n \leq 4$ vision sensors in a sensor network, and the uBot mobile manipulator. The apparatus is being assembled to emulate multiple-task emergency response tasks and to study technologies that allows peer-to-peer training and improves the performance of human-robot teams in the tasks. Assets from the team are deployed to perform: (1) situation assessment; (2) decontamination; (3) evacuation of ambulatory victims; and (4) treatment of non-ambulatory victims. A means of scoring emergency response exercises is used to benchmark the performance of individual agents as well as team performance. The effectiveness of a human-robot team was confirmed in our high level emergency response scenarios. The results suggest that the human-robot team of asymmetric roles can significantly outperform individual agents.

*This work was supported by ONR MURI-N000140710749 and NASA GCT-NNH11ZUA001K.

Contents

1 The E-Response Agent Infrastructure **4**

1.1 Camera Array 4

1.2 uBot-5 Mobile Manipulator 5

1.3 Robot Actions 5

1.3.1 Frontier-Based Search 5

1.3.2 Move-To 6

1.3.3 Pick and Place 6

1.3.4 Localize 6

1.3.5 Emergency Stop 7

1.4 User Interface 7

2 Emergency Response ***Experiments*** **8**

2.1 Subtasks 8

2.2 Scoring 8

2.3 Experimental Controls 10

3 Results **11**

4 Discussion **13**

5 Future Works **13**

5.1 Collaboration Policy Iteration 13

5.2 Activity Models 13

References **14**

Appendices

A Manipulation/Observation Controllers **15**

A.1 Control Objectives 15

B Path Planning **16**

B.1 Potential field based exploration algorithm 17

C Human Tracking **17**

C.1 Triangulate a Human Position in the World Frame 17

C.2 Track a Human 18

D Raw Data from Preliminary Experiments **18**

D.1 Robot-Only Search 18

D.2 Human-Only Mode 21

D.3 Human-Robot Team Mode 24

1 The E-Response Agent Infrastructure

Often, H-R teams require that the human partner, typically the most capable agent, is consumed with situation assessment and tactical decision making for the team. Consequently, the team performance suffers—it can be lower than human performance alone because of the cognitive load on the human member of the team.

Our approach is to examine modes of collaboration that rely on training to yield collaboration policies that unfold without additional deliberation at run-time. In this approach the human partner is on task as much as possible and deliberates only when the current policy needs revision, perhaps when one of the agents reports situations that fall outside of the scope of the current policy.

The Emergency Response (ER) task is a fertile area for research such as this because of a significant asymmetry in the roles of human and robot assets. Humans can be seriously injured or killed in ER operations, so the robot agents can be used to avoid the exposure of human assets. Under these conditions, and with a repertoire of H-R collaboration policies, we expect that the value added of teams including humans and robots to be proven in the short term.

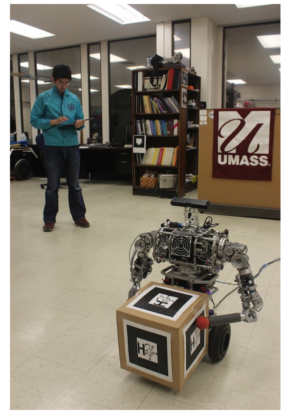
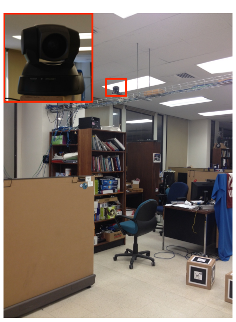


Figure 1.1 *A human-robot team for emergency response*

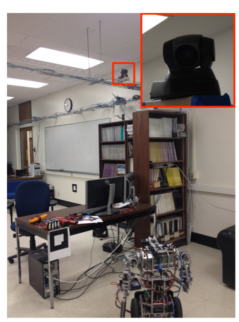
An architecture for exploring these ideas is under development at UMass Amherst in the Laboratory for Perceptual Robotics and this report introduces the approach and some preliminary results (Figure 1.1).

1.1 Camera Array

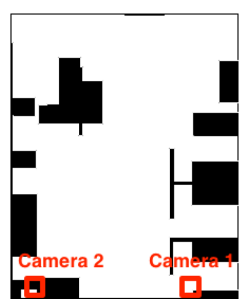
A camera array is deployed to assess the state of the human-robot team (Figure 1.2). We use two fixed Pan-Tilt-Zoom cameras to locate human-peer position. In the real emergency response scenarios, indoor UAVs will be used to gain more sensory information from the scene. In the current settings, our PTZ cameras are fixed and not capable of moving position except controlling pan and tilt angles. Because of this constraint, we limited the size of the space we used in the experiment rather than using the entire room. We want to consider where to deploy UAVs or fixed cameras to gain more information of the scene as a future work.



(a) camera 1

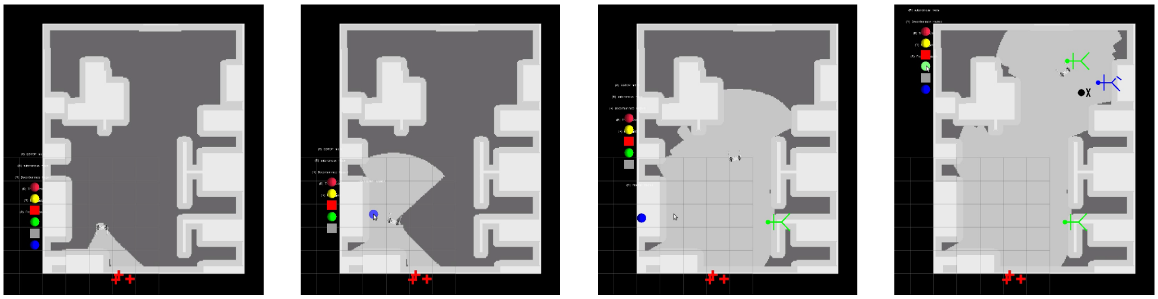


(b) camera 2



(c) sensor array geometry

Figure 1.2 *The sensor array assets in the emergency response system: (a) camera 1; (b) camera 2; and (c) the occupancy grid of the experimental environment with the camera positions marked. Black occupied areas include desks, bookshelves, partitions, and a couch.*



(a) initial map (b) early phase in search (c) map ER targets (d) later phase in search

Figure 1.4 *Progress in the course of a frontier-based search of an autonomous robot aloneasasas*

1.2 uBot-5 Mobile Manipulator

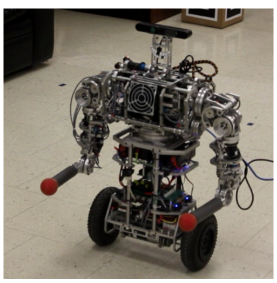


Figure 1.3 *The uBot-5 mobile manipulator.*

The uBot-5 (Figure 1.3) is a small light-weight bimanual mobile manipulator developed at the Laboratory for Perceptual Robotics at UMass Amherst [1]. Each arm has four degrees of freedom (DOF)—two in the shoulder and two in the elbow. In addition, the trunk has one DOF. The uBot is a balancing, differential steering robot that is relatively fast and maneuverable and has a 5 kg bimanual payload capacity. It is equipped with an RGB-D camera with a tilt axis to see the ground in front of the robot. The trunk rotate axis can be used to pan the camera. The firmware and low-level code base is being developed now for uBot-6 [2] and dramatically improved uBot-7 is now being assembled. When 6 and 7 are complete, they can be added to the E-Response team. The robot code base is written in the Robot Operating System (ROS) and will be distributed under an open source software licensing agreement.

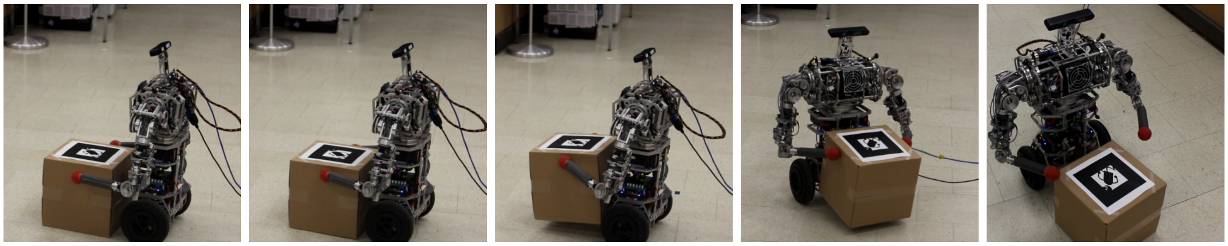
1.3 Robot Actions

The robot is given the following high level actions. The readers who are interested in the details of implementation are referred to the appendix.

1.3.1 Frontier-Based Search

Situation assessment is an important facet of emergency response required to reduce risks to human responders. A frontier-based search method [3] is used to explore unknown areas in the order that optimizes the rate of coverage until it inspects the entire search space. We utilize a harmonic function-based planning algorithm [4] [5] to explore a known map[1]. The search map is a 0.04 m $\times$ 0.04 m grid. In the beginning, the entire search space is marked as a goal (dark gray color in Figure 1.4). The robot continues to consume the unexplored frontier while mapping ER targets until there are no unvisited locations left in the connected component of freespace that includes the robot. The policy can be preempted in response to new state information if appropriate.

[1]We intend to eliminate the need for known maps in future versions.



(a) pre-grasp (b) force closure. (c) lifting (d) navigation (e) release

Figure 1.5 *A sequence of subtasks during the execution of the p-n-p policy*

In Figure 1.4 (c), the robot has located and mapped an ambulatory victim (the green stick figure). According to the trained/programmed policy, the team may elect to respond to this event or to proceed with the frontier-based search. In this example, the latter of the two options is chosen. In Figure 1.4 (d), the search has yielded two ambulatory victims, a non-ambulatory victim (the blue stick figure icon with a broken leg), and has located a hazard target (i.e. a chemical spill) that is indicated by the black skull and crossbones icon.

1.3.2 Move-To

Depending on the state of the search, it may be advantageous to preempt the executing ER policy and respond otherwise. In this case, *Move-To* allows the human partner to change the locale of the robot before resuming the autonomous policy. This action can be used to disperse the robot agents or to separate them from the human responder.

1.3.3 Pick and Place

In laboratory ER exercises, several subtasks make use of pick-and-place (p-n-p) applied to ARcubes (Section 2). Figure 1.5 illustrates the p-n-p action. Given the detected the center position of an ER target, the robot approaches it and determines grasping positions; the robot grasps the target until it achieves force closure; lifts up the target by pre-determined height; navigates to the designated location; and releases the ER target at the location.

1.3.4 Localize

When the response of the ER agent depends on the precision of a ER target location (such as the location of the hazard target since it can harm ambulatory victims and/or human responders that strays too close), the policy (or the human partner) can select the localize action. Selecting this action interrupts the active action and establishes a subgoal to locate a sufficient set of environmental features to localize the robot and/or the target. The localization is done using a suit of known landmarks that are represented by ARtags. The ground truth of these ARtags in the map are known *a priori*.

1.3.5 Emergency Stop

This cancels all executing robot actions. This allows the human responder to halt the robot when, for example, the robot collides with an unmodeled obstacle that it also fails to see.

1.4 User Interface

The experimental platform employs a mixed-initiative control design where the robot selects subtasks according to a policy that is pre-determined. Aggregate state information is supplied to both teammates—the human receives this information through a GUI that also depicts subtasks for both agents as the ER exercise unfolds (Figure 1.6). This interface exposes parts of the coordination policy that may not be effective in some contexts so that the policy can be improved and it allows the human to override the autonomous policy and introduce commands to the robot whenever he/she desires.

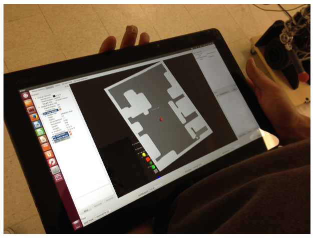


Figure 1.6 *A tablet computer that is used as a graphical user interface*

Skills for addressing ER subtasks are available in the GUI in the form of an icon/command name that follows the robot along the border of the map (Figure 1.4). The human member of the ER team can drag the corresponding command icon and drop it on the map to override the policy and execute the subtask. The place where it is dropped on the map can be used to specify where the subtask should be executed if appropriate. A diagnostic log is maintained when this override option is used for possible future use when refining policies.

2 Emergency Response *Experiments*

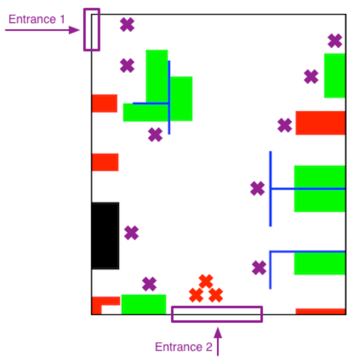


Figure 2.1 *The experiment environment. Agents enter using one of the two possible entrances. Red rectangles are bookshelves, green rectangles are desks, blue lines are partitions, and a black rectangle is a couch. Purple and red crosses show where ER targets are placed for exercises—red crosses indicate medical kits.*

All objects and victims in the laboratory ER exercises we consider are *ARcubes*—28 cm cubes with unique combinations of surface markings. For these initial experiments, four objects are defined by the ARtags presented in Figure 1. The recognition of target objects and useful environmental landmarks rely on Augmented Reality Tags (ARtags) [6]. The uBot-5 autonomously detects and maps ER targets. Multiple observations are fused using a Kalman filter to eliminate false positives and improve precision.

ER trials are executed in a 8.7 m $\times$ 10.8 m space (Figure 2.1). Ten trials are generated by randomly selecting positions for the ARcubes representing 2 hazards, 3 ambulatory victims, and 3 non-ambulatory victims. Hazards are randomly placed in 2 discrete locations. All positions are selected out of 10 candidate positions. The ARcubes that represent 3 medical kits are placed at the same pre-determined positions for each experiment and known to both the robot and the human responder before the exercise begins.

The ER task consists of (1) mapping, (2) decontaminating the environment, (3) evacuating ambulatory victims, and (4) treating non-ambulatory victims within a maximum elapsed time. A coordination policy, the result of practice and the prerogative of the human responder, determine the sequence and priority of each of these tasks as the exercise unfolds.

2.1 Subtasks

The *Treat* subtask is engaged in the neighborhood of a non-ambulatory victim, the ER agent must locate the nearest medical pack and place it next to the victim in question. Doing so constitutes "treating" the victim and accrues the value of this result. *Evacuate* and *decontaminate* subtasks are similar. In the presence of an ambulatory victim, *evacuate* grasps the object that signifies the victim and delivers that object to a safe (pre-designated) location. *Decontaminate* is engaged in reference to a hazard target and the robot moves it to a location designated for hazard containment.

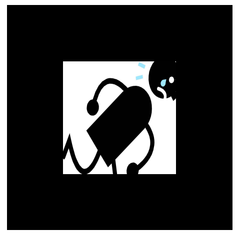
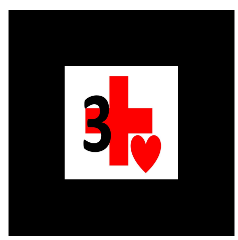
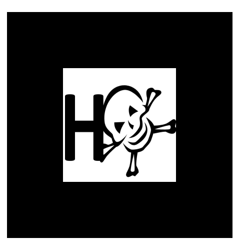
2.2 Scoring

Each ER trial presents an opportunity for the team to accumulate value associated with the collective response. We use the following scores to evaluate the performance.

- treating non-ambulatory victims: if v_{na} is non-ambulatory victim and t is the elapsed time in minutes

$$s_1(v_{na}) = \begin{cases} -100 & \text{if } v_{na} \text{ is not treated} \\ 20 - t & \text{if } v_{na} \text{ is treated} \end{cases}$$

Table 1: Augmented Reality Tags (ARtags) that represent different targets in ER response tasks.

Ambulatory victims			
Non-Ambulatory victims			
Medical kits			
Hazardous materials			

- decontamination: if h is hazard

$$s_2(h) = \begin{cases} 0 & \text{if } h \text{ is not decontaminated} \\ 50 & \text{if } h \text{ is decontaminated} \end{cases}$$

- evacuating ambulatory victims: if v_a is ambulatory victim and t is the elapsed time in minutes

$$s_3(v_a) = \begin{cases} -100 & \text{if } v_a \text{ is not evacuated} \\ 20 - t & \text{if } v_a \text{ is evacuated} \end{cases}$$

- mapping: if n is object, d is the minimum distance (meters) from the position recorded for an object n in the trial to its ground truth position

$$s_4(n) = \begin{cases} 20 - 10d & \text{if } n \text{ is mapped and } d <= 2 \\ 0 & \text{if } n \text{ is not mapped or } d > 2 \end{cases}$$

- death of a human : if Boolean k asserts whether the human responder moved inside a 1.5 m radius of the hazard

$$s_5 = -1000(k)$$

The total score is

$$S = \sum_{v_{na}}^{V_{na}} s_1(v_{na}) + \sum_{h}^{H} s_2(h) + \sum_{v_a}^{V_a} s_3(v_a) + \sum_{n}^{N} s_4(n) + s_5$$

where V_a is a set of ambulatory victims, V_{na} is a set of non-ambulatory victims, H is a set of hazard targets, and $N = V_a + V_{na} + H$.

For the team to perform well, it must exploit asymmetric roles in the human-robot team to acquire value points while avoiding penalties. For example, the robot may address decontamination since the hazard may be fatal to human beings if they venture within the hazard radius. The robot gets points for successfully mapping and decontaminating the space whereas the exposure of the human to the hazard is subject to a stiff penalty.

2.3 Experimental Controls

Robot Only Search The search policy executes a frontier-based search for any of the targets (hazards, victims) until it detects one, at which point it executes the corresponding ER subtask until completion. The robot attempts to localize itself on the map relative to known ARtag landmarks in the environment every 12 seconds if there are landmarks available, else it continues without re-localizing.

Human Only Search The human responder attempts to treat victims while being careful to avoid the presence of hazardous substances that will kill them if they stray within a 1.5 m radius. It is possible for the human responder to be killed in the ER exercise without accruing any other points.

Human-Robot Team Search The robot responds to the emergency using a policy determined *a priori* by the human responder as result of practice exercises. A strawman policy is used in the experiments reported in this paper wherein hazard assessment and decontamination have the highest priority after which the robot completes mapping other ER targets, and evacuates or treats victims. Both agents can add ER targets to the map and announce their intention to address a subtask, thus removing it from the task queue.

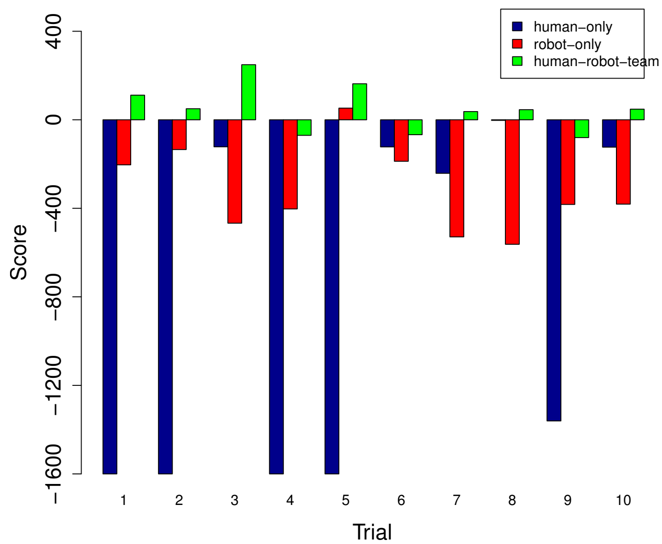


Figure 1: Scores and scores in percentage for all scenarios. The maximum score possible earned by this experiment was 380.

3 Results

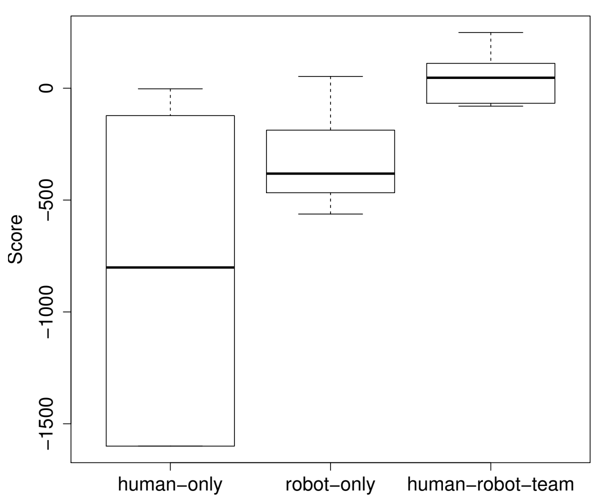


Figure 3.1 *Performance of human-only, robot-only, and human-robot team in condition 1*

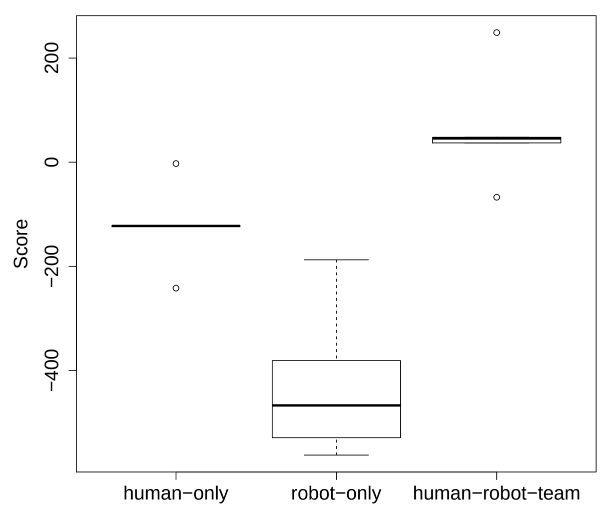


Figure 3.2 *Performance of human-only, robot-only, and human-robot team in condition 2*

Table 2: P values of pairwise comparisons using t-tests with Holm adjustment for condition 1. (* indicates statistical significance at an α level of 0.05)

	human-only	robot-only
robot-only	$p = 0.099$	-
human-robot	$p = 0.010^*$	$p = 0.001^*$

Table 3: P values of pairwise comparisons using t-tests with Holm adjustment for condition 2. (* indicates statistical significance at an α level of 0.05)

	human-only	robot-only
robot-only	$p = 0.038^*$	-
human-robot	$p = 0.042^*$	$p = 0.027^*$

Figure 1 shows the scores for all scenarios. We analyzed the results under two conditions; (1) all trials; (2) trial 1, 2, 4, 5, and 10 where the human responder did not die both in the human-alone and the human-robot team scenario. The performance of the human and that of the robot was not statistically different when we considered the entire trials (Figure 3.1 and Table 2). However, in the condition 2, the human performance was better than that of the robot (Figure 3), which is statistically significant (Table 3). It is interesting to note that the human was able to outperform the robot even though the number of victims the human partner was able to treat was limited by the presence of hazard. This indicates the asymmetric performance of the human and the robot and speaks for the need of asymmetric role assignments. It is also interesting to note that the performance of H-R team was better than that of either of the agents alone in the both conditions. This supports the proposition that the robot is an essential part of ER tasks where the human responders life is in jeopardy and motivates future experiments. This result also corroborates the result of Xie et al. [7] that the H-R team can outperform when the use of the robot assists the human while not overloading him/her cognitively. All these results are supported by the detailed analysis of the data.

On average, the H-R team was able to locate and treat 5 out of 6 victims. The default coordination policy allocated this task exclusively to the human team member while the robot was required to locate hazard targets and decontaminate the search environment. On average, the robot was able to remove 0.6 out of 2 hazard targets in each trial. The fact that the robot was able to successfully decontaminate the hazardous materials directly affected the results of the emergency response performance. For instance, in trial 3, the robot removed both of the hazard targets, which cleared the way for the human partner to complete the treatment of all six victims. In contrast, when the human responder performed alone, they were able to treat only four of these non-ambulatory victims under the same conditions. In trial 2, the robot successfully removed only 1 hazard in the beginning of the episode. However, after this corridor was cleared, the human responder was able to treat five victims—all except one near the remaining hazard.

Even when the robot failed to decontaminate any of the hazards, the human responder had access to the video stream from the robot and to the acquired map. On average, the human responder waited 109.74($\pm$63.55) seconds until executing their first action. When the human did not have a robot teammate, the average time decreases to 10.17($\pm$5.68) seconds. The collaboration policy restrained the human responder from entering until at least one entrance to the environment is cleared. This policy resulted in a 100% survival rate for the human responder. In trials 1 and 2, the human treated 6 and 5 victims, respectively, while staying away from the hazardous materials.

The robot average action took 83.95($\pm$28.33) seconds while the average human action took only 8.34($\pm$4.86) seconds. The human shows significantly higher performance in treating victims. On average, the robot was able to detect/treat victims/hazardous materials with success rate of 27.50%. While, in these experiments, the human responder never failed while treating victims except in the case when he got too close to a hazard and died as result. This suggests that the robot can still contribute to the H-R team performance even if the robot performance alone does not match that of the human.

4 Discussion

The overall robot performance was directly affected by the performance of computer vision and localization. Without any filters, the ARTag detection was not reliable. Throughout the trials, there were a significant number of false positives and false negatives; especially false positives even after filtering. This caused incorrect registration of ER targets on the map and the failure of p-n-p actions. Considering the fact that we used ARTags, the vision performance could be worse if we employ real objects in the same setting. By chance, when the landmark ARTags were not in the view of the uBot-5, it gave up and continued its tasks. The error in odometry did not cause much trouble when the robot was in a large open space. However, when the robot was near obstacles, such as desks or bookshelves, this usually led to undesirable collision that resulted in the early cessation of trials and hence low overall scores.

5 Future Works

5.1 Collaboration Policy Iteration

Currently, the human-robot team search employs a single fixed collaboration policy. Also, the policy is programmed only offline before the start of the experiments. This can be improved by allowing a human responder to give appropriate subtask commands to a robot where it learns the policy that capture the commanded patterns of subtasks during training exercises. When we devise algorithms, we should consider the following factors in order to achieve improvement in the coordination policies.

1. The robot should be able to learn the policies that reflect the human's intention based on a small number of commands. Otherwise, training/programming can be expensive in terms of time.
2. Given the abrupt changes of the human command patterns, the robot should be able to readily reflect the changes. This may require appropriate machine learning algorithms as well as efficient user interfaces.

5.2 Activity Models

In keeping with our emphasis on minimal "off-task" cognitive load for the human responders, we are considering an array of wear-able sensors that can classify human activity (search, decontaminate, evacuate, and treat) based solely on observed activity and using these classes to annotate the aggregate task model and the GUI so that the robot can select better, more complementary tasks and roles.

We have developed a simple system for integrating plan and activity recognition into a single system using Latent Dirichlet Allocation (LDA) topic models. Commonly used for analyzing large quantities of text documents, LDA assumes a "bag-of-words" model where each observation (word) is independent of the others. It learns a distribution of words over topics as well as topics over documents to give us summaries of their essence. In our case, the learned topics represent higher-level actions as distributions of poses observed by a RGB-D sensor and each document, represented by an observation sequence of poses for a single plan execution, has a distribution of these topics which allows us to loosely recognize a plan [8, 9]. Our next step (currently in progress) is to synchronize the recognition system with a plan library so that recognized plans and actions will have a human-interpretable equivalent to the topics that the system learned; this would make it possible to indicate to a robot when it should interact with or assist its human partner.

References

[1] P. Deegan, B. Thibodeau, and R. A. Grupen, "Designing a self-stabilizing robot for dynamic mobile manipulation," in *Proceedings of the RSS Workshop on Manipulation for Human Environments*, Philadelphia, PA, August 16–19 2006.

[2] D. Ruiken, M. Lanighan, and R. Grupen, "Plan and activity recognition from a topic modeling perspective," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Atlanta, GA, October 15–17 2013.

[3] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, July 10–11 1997, pp. 146–151.

[4] C. I. Connolly, J. Burns, and R. Weiss, "Path planning using laplace's equation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, OH, May 13–18 1990, pp. 2102–2106.

[5] C. I. Connolly and R. A. Grupen, "The applications of harmonic functions to robotics," *Journal of Robotic Systems*, vol. 10, no. 7, pp. 931–946, 1993.

[6] "Artoolkit," http://www.hitl.washington.edu/artoolkit/.

[7] D. Xie, Y. Lin, R. Grupen, and A. Hanson, "Intention-based coordination and interface design for human-robot cooperative search," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, September 25–30 2011, pp. 3349–3354.

[8] R. G. Freedman, H. Jung, and S. Zilberstein, "Plan and activity recognition from a topic modeling perspective," in *Proceedings of the 24th International Conference on Automated Planning and Scheduling*, Portsmouth, NH, June 21–26 2014, p. In Press.

[9] R. G. Freedman, H. Jung, R. A. Grupen, and S. Zilberstein, "How robots can recognize activities and plans using topic models," in *Proceedings of the AAAI Workshop on AI and Robotics*, Quebec City, Canada, July 27–31 2014, p. In Press.

[10] S. Hart and R. Grupen, "Learning generalizable control programs," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 3, pp. 216–231, 2010.

[11] Y. Nakamura, *Advanced robotics: redundancy and optimization.* Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1990.

[12] B. Yamauchi, A. Schultz, and W. Adams, "Mobile robot exploration and map-building with continuous localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 16–21 1998, pp. 3715–3720.

[13] itseez, "Opencv," http://www.opencv.org/.

A Manipulation/Observation Controllers

A primitive control action regulates motor response to meet a corresponding sensory goal state. Each controller is defined as $c(\phi, \sigma, \tau)$ where ϕ is a scalar navigation function, σ designates sensory feedback, and τ specifies a set of motor units. Among many possible navigation functions[2], this work incorporates a quadratic potential function

$$\phi(\sigma) = \rho^T \rho \quad \text{where } \rho = \sigma_{ref} - \sigma_{act}.$$

The sensitivity of the potential to changes in the value of motor variables is captured in the Jacobian

$$\mathbf{J} = \frac{\partial \phi(\sigma)}{\partial \tau}.$$

Given this, control signals are computed using the expression

$$\Delta u_\tau = -\kappa \mathbf{J}^{\#} \phi(\cdot)$$

where $\mathbf{J}^{\#}$ is the Moore-Penrose pseudoinverse [11] and $\kappa > 0$. Under this control law, the system follows the negative gradient of the potential toward a stable attractor state where $\nabla_\tau \phi(\cdot) = 0$ to relate σ_{ref} and σ_{act} as a closed-loop circuit.

A control action may be configured to address multiple objectives simultaneously. These objectives are specified by multiple primitive control actions composed using nullspace projections that reflect the order of priority. Consider a composite controller that is constructed from controllers c_1 and c_2 where c_1 is "superior" controller and c_2 is the subordinate controller. They are represented in a single composite controller, using

$$\Delta u_\tau = -\kappa_1 \mathbf{J}_1^{\#} \phi_1(\cdot) - \mathcal{N}_1 \left(\kappa_2 \mathbf{J}_2^{\#} \phi_2(\cdot) \right)$$

where $\mathcal{N}_1$ represents the nullspace of the higher priority controller, $\mathcal{N}_1 = I - J_1^{\#} J_1$. This ensures that a lower priority objective does not destructively interfere with the progress of the higher priority objective. Nullspace projections can be cascaded to incorporate more than two simultaneous objectives, however only the highest priority objective is guaranteed to reach a fixed point [11].

A.1 Control Objectives

Endpoint Position Controller The endpoint position controller (C_{EP}) minimizes the error between the goal state and the current actual state by

$$\mathbf{e} = \mathbf{x}_{ref} - \mathbf{x}_{act}$$

which allows the operator to control the Cartesian position of both hands explicitly.

Endpoint Force Controller Since the uBot-6 is not equipped with tactile force sensors, we approximate force feedback using a linear spring system defined on the contact point of the uBot-6's each hand. Ideally, this force should be applied along the surface normal of the contact point. However, in this work, it is approximated by the direction to the other hand from the hand being controlled, which is governed by C_O. A linear spring is in the form of

$$f = k \Delta x$$

[2] Alternatives can be found in Hart and Grupen [10].

where $k \in R^+$ is a spring constant representing the characteristic stiffness of the object; we use $k = 1$ throughout this work. Δx is the amount of spring displacement occurred from its relaxed position. In order to maintain the same magnitude of force f, the end effector needs to press the object by Δx. This is achieved by minimizing

$$\mathbf{e} = \Delta\mathbf{x}_{ref} - \Delta\mathbf{x}^*_{act}$$

where

$$\Delta\mathbf{x}^*_{act} = \mathbf{x}_{des} - \mathbf{x}^*_{act},$$

$$\mathbf{x}^*_{act} = \frac{\mathbf{x}_{act} \cdot (\mathbf{x}'_{ref} - \mathbf{x}_{act})}{|\mathbf{x}'_{ref} - \mathbf{x}_{act}|^2}(\mathbf{x}'_{ref} - \mathbf{x}_{act})$$

and $\mathbf{x}^*_{act}$ is $\mathbf{x}_{act}$ projected on the vector $\mathbf{x}'_{ref} - \mathbf{x}_{act}$. As in C_O, $\Delta\mathbf{x}_{ref}$ is set to $\Delta\mathbf{x}^*_{act}$ at the mode switch to the *Force mode by the teleoperator.*

Object Position Controller *The controller (C_{OP}) regulates the Cartesian position of the object in the robot's hands and indirectly controls the robot's hands by*

$$\mathbf{e} = \mathbf{x}_{ref} - \mathbf{x}_{act}$$

where $\mathbf{x_{ref}}$ is the target Cartesian position of the object. It is assumed that the object is at the center of the two hands and computed by

$$\mathbf{x}_{act} = \frac{\mathbf{x}_{act} + \mathbf{x}'_{act}}{2}$$

where $\mathbf{x}_{act}$ is the current position of the hand being controlled and $\mathbf{x}'_{act}$ is the other.

Head Controller *The head controller (C_H) controls the rotational joints θ in the head by*

$$\mathbf{e} = \theta_{ref} - \theta_{act}.$$

In the Unaided mode, θ_{ref} is controlled directly by the operator while, in the Head mode, it is controlled by the teleoperator to gaze upon the center of the both hands.

B Path Planning

We employed harmonic functions to conduct a potential field path planning [4] [5]. A harmonic function $\phi(x, y)$ on a domain $\Omega \subset R^n$ satisfies the Laplace's equations:

$$\nabla^2\phi = \sum_{i=1}^{n} \frac{\partial\phi}{\partial x_i^2}$$

The Dirichlet boundary conditions are used:

$$phi|_{\partial\Omega} = c,$$

where $\partial\Omega$ represents a boundary of a map. c is set to be 1 in our case. This means that all grids on the boundary or the obstacles in a potential field are always set to be 1. Potentials on goal region are set to be zero.

$$\phi|_{goal\ region} = 0$$

We use Successive Over Relaxation(SOR) to compute numerical solutions $u(x_i, y_j)$ of Laplace's equation. SOR on Laplace's equation is:

$$\begin{aligned} u^{(k+1)}(x_i, y_j) &= u^{(k)}(x_i, y_j) + \frac{\omega}{4}(u^{(k+1)}(x_{i+1}, y_j) + \\ &\quad u^{(k+1)}(x_{i-1}, y_j) + u^{(k)}(x_i, y_{j+1}) + \\ &\quad u^{(k)}(x_i, y_{j-1}) - 4u^{(k)}(x_i, y_j)), \end{aligned}$$

where k is the iteration number, ω is the acceleration constant, (x_i, y_j) represents positions in a occupancy grid map. We skip the computation of $u^{(k+1)}(x_i, y_j)$ if (x_i, y_j) is on the boundary, the obstacles or goal regions.

We compute a path from the current ubot position to the goal region by following the gradient of the potential field, $\nabla\phi$. We use the following equations to obtain the numerical solutions of $\nabla\phi$.

$$\begin{aligned} \frac{\partial u(x_i, y_j)}{\partial x} &= \frac{u(x_i - \frac{d}{2}, y_j) - u(x_i + \frac{d}{2}, y_j)}{d} \\ \frac{\partial u(x_i, y_j)}{\partial y} &= \frac{u(x_i, y_j - \frac{d}{2}) - u(x_i, y_j + \frac{d}{2})}{d} \\ \nabla u(x_i, y_j) &= \arctan\left(\frac{\partial u(x_i, y_j)}{\partial x}, \frac{\partial u(x_i, y_j)}{\partial y}\right), \end{aligned}$$

where d is the length of a cell in an occupancy map.

B.1 Potential field based exploration algorithm

We utilize the potential filed path planning algorithm we described above to explore a map that is given prior to the exploration. The key idea is similar to the frontier search method [12] in the sense that our algorithm also navigates the robot to one of the grid cells on frontiers. $\phi(x, y)$ is goal region if $\phi(x, y)$ is unexplored region and on neither boundaries nor obstacles. Then, the gradient of the potential map leads to one of the goal grid on frontiers. The robot keeps updating explored area and frontiers even while the robot is moving. The path will be updated before it reaches the original goal on frontiers if the potential field changes. The robot keeps searching the map until every reachable cell is explored.

C Human Tracking

C.1 Triangulate a Human Position in the World Frame

We use two PTZ cameras to triangulate a human position. We use calcBackProject function and camshift function in OpenCV [13]. In each image obtained form cameras, we first calculate back projection of a *histogram model of a human. In our experiment, a human wears a bright colored jacket. We use the histogram of this jacket as a histogram model. Then, we apply camshift to find the window position, size and rotation of the target in the image. We consider the center of the window as the target position in the image. Kalman filter runs on the target position in the image to make the tracking system more robust. In order to triangulate the target position in the world frame, we first compute a ray from a camera origin through the target position for each camera image. A ray from a camera k is :*

$$\mathbf{v_k} = \begin{bmatrix} \frac{(x_k - p_{13}^k)}{p_{11}^k} \\ \frac{(y_k - p_{23}^k)}{p_{22}^k} \\ 1.0 \\ 0 \end{bmatrix}^{\mathbf{C^k}},$$

where p_{ij}^k is an element of a projection matrix of a camera k, C^k is a frame of camera k and (x_k, y_k) is the position of the target in the image from a camera k. The ray in the world frame is :

$$D_k \quad = \quad {}_W T_{C^k} \mathbf{v_k} - \mathbf{P_k}$$

,where P_k is a fourth column of ${}_W T_{C^k}$. ${}_W T_{C^k}$ is a transformation matrix from a camera frame k to the world frame W. Let D_m be the ray from a camera m in the world frame. We compute the closest point to these two lines, $l_k = P_k + sD_k$ and $l_m = P_m + tD_m$. Only when the minimum distance between these two lines is below the threshold, we use this closest point as a target position.

C.2 Track a Human

Let (c_x, c_y) be a center of an image from a camera k. We compute a distance from the target position in the image to the center of the image (x, y).

$$d_x = c_x - x$$

$$d_y = c_y - y$$

A camera controls its pan and tilt angle until $|d_x| < a$ and $|d_y| < b$ where a and b are thresholds.

D Raw Data from Preliminary Experiments

Throughout the section, AV, NAV, HM represent ambulatory victim, non-ambulatory victim, hazardous material respectively.

D.1 Robot-Only Search

Scenario 1

1. *Start treating AV at 24.98*
2. *Finish treating AV at 1:51.21*
3. *Start treating AV at 3:32.52*

4. *Finish treating AV at 4:03.90*
5. *Start treating NAV at 4:03.94*
6. *Finish treating NAV at 4:47.02*

The uBot-5 failed to complete the task since it stepped into the occupied grid of the map. Total runtime is 7:46.77. The total count of successfully treated victims is 3. The average time for treatment is 53.56($\pm$28.89) seconds.

Scenario 2

1. *Start treating NAV at 1:26.51*
2. *Finish treating NAV at 2:38.15*
3. *Start treating HM at 2:42.69*
4. *Finish treating HM at 3:52.31*
5. *Start treating NAV at 3:52.34*
6. *Finish treating NAV at 4:39.82*
7. *Start treating NAV at 5:11.02*
8. *Finish treating NAV at 6:46.02*

Total runtime is 6:46.02. The total count of successfully treated victims is 3 and that of decontaminated hazardous material is 1. The average time for treatment is 70.94($\pm$19.42) seconds.

Scenario 3

1. *Start treating NAV at 1:14.01*
2. *Finish treating NAV at 2:34.95*

The uBot-5 ran only about 3 mins because it stepped into the occupied grid of the map and failed to recover. Total runtime is 2:34.95. The total count of successfully treated victims is 1, which took 80.94 seconds.

Scenario 4

1. *Start treating AV at 29.86*
2. *Finish treating AV at 2:28.11*
3. *Start treating HM at 2:28.14*
4. *Finish treating HM at 3:39.36*

Total runtime is 3:39.36. The total count of successfully treated victims is 1 and that of decontaminated hazardous material is 1. The average time for treatment is 94.74(±33.26) seconds.

Scenario 5

1. *Start treating NAV at 43.54*
2. *Finish treating NAV at 1:54.76*
3. *Start treating AV at 1:54.79*
4. *Finish treating AV at 4:37.74*
5. *Start treating AV at 4:42.28*
6. *Finish treating AV at 6:11.24*
7. *Start treating NAV at 6:11.27*
8. *Finish treating NAV at 7:11.62*
9. *Start treating AV at 7:34.66*
10. *Finish treating AV at 9:04.73*

Total runtime is 10:00.00. The total count of successfully treated victims is 5. The average time for treatment is 94.71(±40.14) seconds.

Scenario 6

1. *Start treating AV at 15.01*
2. *Finish treating AV at 2:16.41*
3. *Start treating NAV at 2:16.44*
4. *Finish treating NAV at 3:17.89*
5. *Start treating HM at 5:52.42*
6. *Finish treating HM at 7:34.92*
7. *Start treating HM at 8:03.03*
8. *Finish treating HM at 9:55.76*

Total runtime is 10:00.00. The total count of successfully treated victims is 2 and that of decontaminated hazardous material is 2. The average time for treatment is 99.52(±26.53) seconds.

Scenario 7

1. *Start treating HM at 12.97*
2. *Finish treating HM at 1:35.89*

Total runtime is 5:33.00. The total count of decontaminated hazardous material is 1, which took 82.92 seconds.

Scenario 8

Total runtime is 2:41.51.

Scenario 9

1. *Start treating NAV at 4.97*
2. *Finish treating NAV at 1:56.99*
3. *Start treating HM at 5:34.64*
4. *Finish treating HM at 6:15.23*

Total runtime is 10:00.00. The total count of successfully treated victims is 1 and that of decontaminated hazardous material is 1. The average time for treatment is 76.31(±50.51) seconds.

Scenario 10

1. *Start treating AV at 1:57.68*
2. *Finish treating AV at 3:28.69*
3. *Start treating NAV at 4:27.63*
4. *Finish treating NAV at 5:40.34*

Total runtime is 10:00.00. The total count of successfully treated victims is 2. The average time for treatment is 81.85(±12.96) seconds.

D.2 Human-Only Mode

Scenario 1

1. *Dead at 1.12*

Total runtime is 00:01.12.

Scenario 2

1. *Dead at 1.53*

Total runtime is 00:01.53.

Scenario 3

1. Start treating AV at 6.96
2. Finish treating AV at 22.63
3. Start treating NAV at 22.89
4. Finish treating NAV at 26.33
5. Start treating NAV at 30.94
6. Finish treating NAV 34.21
7. Start treating AV at 36.63
8. Finish treating AV at 40.19

Total runtime is 1:14.24. The total count of successfully treated victims is 4. The average time for treatment is 6.49(±6.12) seconds.

Scenario 4

1. Dead at 9.67

Total runtime is 00:09.67.

Scenario 5

1. Start treating NAV at 15.85
2. Dead at 31.73

Total runtime is 00:31.73.

Scenario 6

1. Start treating AV at 12.65
2. Finish treating AV at 14.97
3. Start treating AV at 21.66
4. Finish treating AV at 24.62
5. Start treating NAV at 34.51
6. Finish treating NAV at 47.41
7. Start treating AV at 48.17
8. Finish treating AV at 59.42

Total runtime is 1:22.39. The total count of successfully treated victims is 4. The average time for treatment is 9.61(±4.91) seconds.

Scenario 7

1. *Start treating AV at 15.40*
2. *Finish treating AV at 22.19*
3. *Start treating NAV at 23.59*
4. *Finish treating NAV at 28.46*
5. *Start treating AV at 47.26*
6. *Finish treating AV at 1:03.91*

Total runtime is 1:04.39. The total count of successfully treated victims is 3. The average time for treatment is 9.44(±6.32) seconds.

Scenario 8

1. *Start treating NAV at 8.51*
2. *Finish treating NAV at 11.17*
3. *Start treating AV at 15.35*
4. *Finish treating AV at 18.84*
5. *Start treating NAV at 21.23*
6. *Finish treating NAV at 25.92*
7. *Start treating AV at 35.60*
8. *Finish treating AV at 44.16*
9. *Start treating NAV at 46.48*
10. *Finish treating NAV at 55.02*

Total runtime is 59.33. The total count of successfully treated victims is 5. The average time for treatment is 5.59(±2.80) seconds.

Scenario 9

1. *Start treating AV at 13.02*
2. *Finish treating AV at 16.57*
3. *Start treating NAV at 22.39*
4. *Finish treating AV at 25.09*

5. *Dead at 35.64*

Total runtime is 35.64. The total count of successfully treated victims is 2. The average time for treatment is 3.13(±0.60) seconds.

Scenario 10

1. *Start treating AV at 17.02*
2. *Finish treating AV at 22.04*
3. *Start treating AV at 30.34*
4. *Finish treating AV at 37.57*
5. *Start treating AV at 52.54*
6. *Finish treating AV at 1:02.48*
7. *Start treating NAV at 1:20.08*
8. *Finish treating NAV at 1:30.62*

Total runtime is 1:48.68. The total count of successfully treated victims is 4. The average time for treatment is 8.13(±2.55) seconds.

D.3 Human-Robot Team Mode

Scenario 1

1. *Start treating AV at 57.78*
2. *Finish treating AV at 1:05.18*
3. *Start treating AV at 1:06.32*
4. *Finish treating AV at 1:10.72*
5. *Start treating NAV at 1:14.72*
6. *Finish treating NAV at 1:20.17*
7. *Start treating NAV at 1:24.33*
8. *Finish treating NAV at 1:31.13*
9. *Start treating AV at 1:35.81*
10. *Finish treating AV at 1:43.44*
11. *Start treating NAV at 1:46.95*
12. *Finish treating NAV at 1:56.31*

Total runtime is 2:45.58. The total count of successfully treated victims by the human peer is 6. The average human time for treatment is 6.84($\pm$1.74) seconds.

Scenario 2

1. *Start treating HM at 1:15.13*
2. *Finish treating HM at 2:45.33*
3. *Start treating NAV at 2:46.28*
4. *Finish treating NAV at 2:50.93*
5. *Start treating AV at 2:56.67*
6. *Finish treating AV at 3:04.87*
7. *Start treating AV at 3:07.69*
8. *Finish treating AV at 3:12.66*
9. *Start treating NAV at 3:14.79*
10. *Finish treating NAV at 3:24.47*
11. *Start treating AV at 3:29.76*
12. *Finish treating AV at 3:42.99*
13. *Register HM at 3:53.48*

Total runtime is 7:11.63. The total count of successfully treated victims by the human peer is 5. The average human time for treatment is 8.15($\pm$3.55) seconds. The total count of decontaminated hazardous material is 1, which took 90.2 seconds.

Scenario 3

1. *Start treating HM at 1:14.01*
2. *Finish treating HM at 2:46.24*
3. *Start treating NAV at 2:46.91*
4. *Finish treating NAV at 2:52.26*
5. *Start treating AV at 3:00.95*
6. *Finish treating AV at 3:05.02*
7. *Start treating NAV at 3:06.39*
8. *Finish treating NAV at 3:11.58*
9. *Start treating HM at 3:30.34*
10. *Finish treating HM at 5:08.99*

11. *Start treating AV at 3:37.54*
12. *Finish treating AV at 3:45.84*
13. *Start treating AV at 3:59.27*
14. *Finish treating AV at 4:06.98*
15. *Start treating NAV at 4:10.63*
16. *Finish treating NAV at 4:19.47*

Total runtime is 4:19.47. The total count of successfully treated victims by the human peer is 6. The average human time for treatment is 6.58(±1.95) seconds. The total count of decontaminated hazardous material is 2. The average human time for treatment is 95.44(±4.54) seconds.

Scenario 4

1. *Start treating AV at 3:47.12*
2. *Finish treating AV at 3:49.54*
3. *Start treating NAV at 3:51.85*
4. *Finish treating NAV at 3:58.35*
5. *Start treating AV at 4:13.22*
6. *Finish treating AV at 4:23.33*
7. *Start treating AV at 4:30.04*
8. *Finish treating AV at 4:40.29*

Total runtime is 4:40.29. The total count of successfully treated victims by the human peer is 4. The average human time for treatment is 7.32(±3.70) seconds.

Scenario 5

1. *Start treating NAV at 56.00*
2. *Finish treating NAV at 57.98*
3. *Start treating AV at 1:00.95*
4. *Finish treating AV at 1:06.26*
5. *Start treating AV at 3:07.73*
6. *Finish treating AV at 3:23.48*
7. *Start treating NAV at 3:26.33*
8. *Finish treating NAV at 3:40.76*
9. *Start treating AV at 3:43.76*

10. *Finish treating AV at 3:54.11*
11. *Start treating NAV at 3:56.98*
12. *Finish treating NAV at 4:11.98*

Total runtime is 5:22.43. The total count of successfully treated victims by the human peer is 6. The average human time for treatment is 10.47(±5.71) seconds.

Scenario 6

1. *Start treating NAV at 2:25.10*
2. *Finish treating NAV at 2:27.79*
3. *Start treating AV at 3:37.78*
4. *Finish treating AV at 3:54.77*
5. *Start treating AV at 3:58.28*
6. *Finish treating AV at 4:25.77*
7. *Start treating NAV at 5:15.65*
8. *Finish treating NAV at 5:26.37*

Total runtime is 5:26.37. The total count of successfully treated victims by the human peer is 4. The average human time for treatment is 14.47(±10.47) seconds.

Scenario 7

1. *Start treating NAV at 1:59.96*
2. *Finish treating NAV at 2:10.55*
3. *Start treating AV at 2:11.26*
4. *Finish treating AV at 2:17.96*
5. *Start treating AV at 2:40.78*
6. *Finish treating AV at 2:56.35*

Total runtime is 7:36.42. The total count of successfully treated victims by the human peer is 3. The average human time for treatment is 10.95(±4.45) seconds.

Scenario 8

1. *Start treating NAV at 38.56*
2. *Finish treating NAV at 42.96*

3. *Start treating AV at 46.70*
4. *Finish treating AV at 55.24*
5. *Start treating NAV at 56.59*
6. *Finish treating NAV at 1:05.02*
7. *Start treating AV at 1:22.28*
8. *Finish treating AV at 1:32.84*
9. *Start treating NAV at 1:37.23*
10. *Finish treating NAV at 1:47.11*

Total runtime is 3:09.61. The total count of successfully treated victims by the human peer is 5. The average human time for treatment is 8.36(±2.39) seconds.

Scenario 9

1. *Start treating AV at 3:19.50*
2. *Finish treating AV at 3:20.20*
3. *Start treating NAV at 3:28.50*
4. *Finish treating NAV at 3:33.23*
5. *Start treating AV at 3:42.88*
6. *Finish treating AV at 3:58.63*
7. *Start treating AV at 4:09.27*
8. *Finish treating AV at 4:23.62*

Total runtime is 4:27.59. The total count of successfully treated victims by the human peer is 5. The average human time for treatment is 8.88(±7.33) seconds.

Scenario 10

1. *Start treating NAV at 1:44.25*
2. *Finish treating NAV at 1:47.72*
3. *Start treating AV at 1:58.56*
4. *Finish treating AV at 2:04.24*
5. *Start treating NAV at 2:41.07*
6. *Finish treating NAV at 2:52.19*
7. *Start treating AV at 2:57.76*

8. Finish treating AV at 3:13.65

9. Start treating AV at 3:23.53

10. Finish treating AV at 3:30.40

Total runtime is 4:02.22. The total count of successfully treated victims by the human peer is 5. The average human time for treatment is 8.61($\pm$4.93) seconds.