In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*
New Orleans, Louisiana, USA, April-May 2004

Directional Audio Beacon Deployment: an Assistive Multi-Robot Application

Dylan A. Shell and Maja J Matarić
Interaction Laboratory
Center for Robotics and Embedded Systems
Department of Computer Science, University of Southern California
Los Angeles, CA USA 90089-0781
{shell|mataric}@usc.edu

***Abstract*— This paper addresses the problem of directional audio beacon deployment. We describe how these beacons can be used on mobile robots to produce a system that can self-deploy and aid in disaster recovery efforts. A distributed algorithm that uses explicit communication to coordinate the deployment process is presented. The algorithm employs existing multi-robot task allocation methodologies and a procedure for clustering potential deployment locations in a problem domain-specific manner. Results from a sensor-based multi-robot simulation demonstrate that self-deploying beacons are indeed feasible and have the potential to decrease expected egress time. Furthermore, we show that the implementation is free of simulator-specific anomalies through trials with a group of physical robots.**

I. Introduction

It has been widely recognized that robotics may have a vital role to play in future disaster recovery efforts. Current robotics work concentrates on search and rescue tasks that require robots overcome highly uncertain environments in order to locate, treat and possibly transport those who are incapacitated [3]. In contrast, the focus of this paper is in assisting those who are already capable of evacuating themselves.

We describe the recent development of audio beacons intended to direct evacuees toward exits efficiently. These beacons produce a sound easily identified and localized by the human auditory system and thus, they are able to assist those who are seeking a route to safety. Beacons must be appropriately positioned in the environment if evacuees are to be guided effectively. Consequently, we study the problem of autonomously deploying a network of these audio beacons using a team of robots. This paper describes a suitable distributed algorithm that is robust with respect to both single node failures and a class of communication failures. Using a simple model, we demonstrate that the expected egress time is favorably affected by the deployed beacons, and further, that even a small number of beacons can be surprisingly effective in facilitating evacuation.

Repeated trials in a multi-robot simulator confirm that our implementation performs appropriately: robots select and navigate toward suitable locations while simultaneously ensuring that their final destinations are not unnecessarily close to one another. Also, any robots that are subsequently added are utilized in an effective manner. Finally, we also validated the algorithm on a collection of physical mobile robots, illustrating that the implementation is robust with respect to physical sensor noise and real world uncertainty.

Fig. 1. Directional audio beacons. (Photos courtesy of Sound Alert Technology PLC, Brigade PLC & Klaxon Signals.)

II. Directional Sound Beacons

Visual cues are people's primary means for spatial localization, and consequently there is the potential to become disoriented or possibly even lose one's way in low visibility conditions. The most tragic examples are of deaths that occur because victims are unable to locate suitable emergency exits in time. See, for example, the Dusseldorf Airport Disaster [5] in which asphyxiated bodies were discovered less than three meters from an emergency exit.

One solution to these problems is the use of directional audio beacons [15]. These beacons (see Figure 1) emit a sound in a range of frequencies known to be well-suited for localization by the human ear. Experiments indicate that people can be directed through a complex navigation task if audio beacons are placed at strategic points en route. Multi-floored buildings permit beacons to play an enhanced role. In addition to the standard pulsing broadband sound, those beacons near stairwells can play a sweeping melodic tone, ascending tonality indicating that subjects should travel upward, descending tonality indicated the reverse. In one experiment, the subjects were not informed about the meaning of these sounds, nevertheless not a single participant in any of the trials mistakenly took a wrong turn or accidentally ended up in an incorrect room [15].

Trials conducted with human participants (both with and without decreased visibility) overwhelmingly established the effectiveness of the beacons in aiding evacuation. For instance, in one trial, people got lost in an environment with 100% visibility. They had successfully evacuated the same environment, while smoke-filled, only minutes before with the aid of audio beacons [15].

Sound beacons have other implications, including: diminishing band-wagon effects, heightening the sense of urgency, possibly reducing congestion, and limiting time spent moving toward non-existent exits. These factors also decrease the risks that emergency personnel must endure.

III. THE PROBLEM

While directional audio beacons have been shown to be effective, all evaluations to date have made the assumption that the beacons are *already* strategically situated in the environment. Currently, buildings with existing audio beacon installations are the exception rather than the rule, thus we propose that robots be used to automatically deploy these beacons as needed. Two different solutions to this problem, each inspired by existing work in sensor network deployment, present themselves. The first involves some number of robots depositing beacons as they move through their environment (see for example the sensor network deployment strategy in Batalin and Sukhatme [1]). The second provides each robot with a beacon that can be activated when the robot arrives at an appropriate location. The remainder of this paper considers only this second case, where the beacons can be said to self-deploy.

The purpose of this work is to propose, present and validate a distributed coordination algorithm for audio beacon self-deployment; the following assumptions are deemed reasonable in light of this goal.

A. Assumptions

The robots are assumed to be provided with a map of the environment into which they will be deployed. The map consists of both the metric information (needed for localization) and a topological graph. The latter representation could be automatically generated from the former (as in Thrun and Bucken [12] for example); in our implementation it is explicitly provided.

Since the building to be evacuated may have more than a single level, we employ a map for each floor. A list of links between the floors (at the locations that represent stairwells) and any supplementary information, such as the locations of fire exits, must be provided in addition to the map. What results is essentially a $2\frac{1}{2}$ dimensional representation for the multi-level building. See Figure 2 as an example of a map for a building with two floors.

Each robot is provided with a rough estimate of its initial location enabling it to quickly procure a unimodal pose estimate with high certainty. Our algorithm can be extended to have an initial localization phase, but that is not directly relevant to the deployment problem.

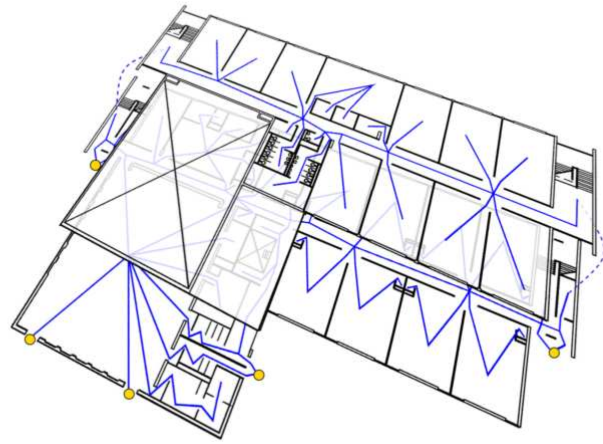

Fig. 2. The representation of a multi-fbor environment used for deployment of the beacons. The yellow dots mark the emergency exits, note the dotted line segments indicate the stairway connections in the planar topological overlay. Assignment considers where the robots are currently positioned ensuring that the distance traveled is minimized wherever possible.

No serious navigational hindrances are anticipated; this corresponds to the case of a smoke-filled, but otherwise perfect building. The robots need not deal with collapsed structures, or gross inconsistencies between the map and the environment. Nevertheless, a local navigation mechanism that is capable of avoiding large obstacles (e.g., other robots, water coolers, debris, etc.) is employed.

Finally, it is assumed that complex acoustic effects (multipath, interference, etc.) need not be considered when calculating the suitability of a particular location for beacon deployment. This is reasonable because the beacons are of a sophisticated design specifically intended to avoid these effects.

B. Unique Properties of the Problem Domain

Numerous deployment algorithms exist, but this particular domain presents a number of new challenges. Two major ones are covered here.

Typical deployment algorithms intended for sensor networks tend to focus on the notion of coverage, i.e., maximizing the total area collectively sensed. In the evacuation scenario, coverage is less important than providing useful information about the location of the exits. No formal function describes the suitability of a particular location (or set of locations) in terms of expected life-saving applicability. For example, Withington [15] used fire exits and stairwells as assumed strategic locations[1].

Secondly, the effectiveness of the deployment algorithm is difficult to evaluate empirically. This property is shared with other human-robot interaction problem domains. Performing extensive experiments with humans is not feasible, and capturing a truly realistic evacuation scenario is a practical impossibility. Section V gives more details of regarding how tackled this problem was tackled.

[1]Personal Correspondence, Oct 16, 2003.

IV. The Approach

In order to make use of existing work in the area of multi-robot task allocation (MRTA), we consider the deployment task to be an instantaneous assignment problem with single-task robots and single-robot tasks. It has been shown that this particular class of MRTA problems can be treated as an Optimal Assignment Problem (OAP) [7] with a known tractable solution. However, casting the problem into the MRTA framework in a naïve manner results in inappropriate assignments: this is demonstrated, next.

A direct way to cast the deployment problem into a MRTA instance, is to consider the potential deployment locations as tasks for each of the robots. This requires the construction of a utility matrix, each element therein is an estimate of the utility, or expected worth, of a particular assignment. The $(i, j)^{\text{th}}$ utility matrix entry is for robot i being deployed to location j, and can be considered the *reward* for having a beacon at location j, less the *cost* expected to be incurred by that robot during navigation to location j. Once the full matrix has been constructed, robots can be assigned to tasks.

Unfortunately, the OAP formalism requires fixed non-interrelated utilities, a requirement that is not satisfied by this example. The reward obtained by having a beacon at a particular location depends not only on the environmental constraints, but also on where the other beacons are positioned. Imagine that one exit is in an excellent location; assigning two robots close to that exit is inefficient, because once a single robot has been deployed to a location the value of having others nearby is greatly lessened.

To avoid this problem we permit a task (in the MRTA framework) to include a set of locations within some local neighborhood; a robot being assigned to that task is ensured that no other robots will be deployed within that neighborhood. Next, we discuss the problem in three phases: 1) choosing a list locations suitable for deployment and clustering them into tasks, 2) assigning robots to those tasks, and 3) having robots each navigate toward their appropriate locations. Each of these is discussed in turn. The final subsection presents the deployment algorithm itself in greater depth.

A. Where to deploy

In addition to the simplistic utility estimate calculation discussed above, a fully accurate estimate would need to account for the uncertainty in the robot's pose estimate and various other interference effects that could occur during navigation. Pose uncertainty is ignored because, as stated in Section III-A, the the robot is provided with a good estimate of its initial location and a metric map, these permit accurate location tracking. Robotic traffic congestion problems are ignored because the number of robots used is small when compared with the size of the environment. Furthermore, the robots will purposefully spread out over time, so congestion is likely to be minimal.

Feasible locations are selected from the set of emergency exits and stairways. They are then clustered based on distance from one another. Assigning robots to entire clusters ensures that no other robot will be allocated a location within the cluster, thus separating deployed beacons by a minimum threshold distance.

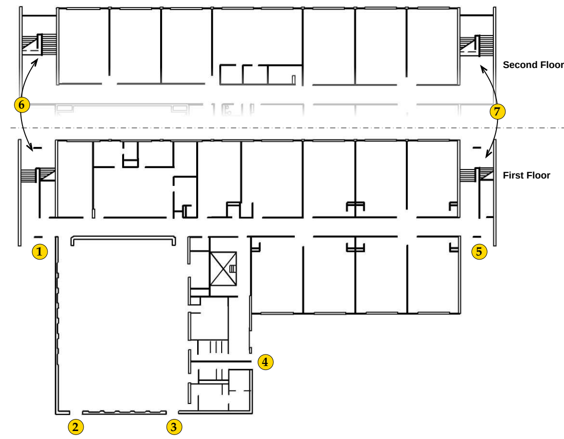

Fig. 3. The test environment with the emergency exits marked 1–5 and the connection stairways 6 and 7. The second fbor is only partially shown, but can be seen in Figure 2.

Consider, for example, the two-floor environment shown in Figure 3. When deploying a single robot we consider the partition of feasible locations to be the set of all possible destinations, $S_{0,1} = \{1, ..., 7\}$. If another robot is added, then the environment is partitioned into two sets of locations, $S_{0,2} = \{1, ...4, 6\}$ and $S_{1,2} = \{5, 7\}$. This is performed using an approach similar to Krushkal's Algorithm [4] for construction of a minimal spanning tree (MST). The last step of Krushkal's algorithm merges the two sets of vertices, $S_{0,2}$ and $S_{1,2}$, to form $S_{0,1}$. Rather than constructing the full MST, the last merger need not be performed, resulting in a forest of size two. More generally, the set merger step is applied only until the correct number of tasks remains; the length of the last edge added to the MST is effectively the cluster threshold.

The utility estimate for a cluster is calculated as the maximum utility over all the members of that cluster. All distances used in the calculations are based on actual distances involved in navigating between locations; they incorporate the known obstacles.

A further complication arises since one requires not only that there be a sufficient number of tasks, but also that there exist a feasible assignment of tasks to robots. This is easily handled by calculating the sets as described above, thereafter relaxing the threshold until a feasible assignment is found for all robots; if none exits for a particular robot, then that robot is assigned no task, i.e., a *no-op* task.

B. Whom to deploy

Once the set of tasks have been generated, the robots are assigned to those tasks. Gerkey and Matarić [7] propose the use of the Hungarian Method for calculating these assignments; in our implementation the calculation of utility matrix entries was far more computationally intensive than actually performing the assignment.

Because robots are assumed to be constrained to a particular floor, the resulting utility matrix has a particular structure: two robots on adjacent floors have positive utility estimates for destinations on their own floor and zero for the locations elsewhere, stairways between the floors are the only cases where both robots may have non-zero utilities. Stairwells are effectively locations that exist on both floors – if a robot is tasked with one of these locations it moves to either the head or foot of the stairwell as appropriate.

If we were to ignore the existence of stairs (and assume that sound did not permeate between floors) then the assignment problems could be solved independently on each floor. We do not do this because stairwells are extremely important during evacuation. Most exits occur on the ground floor, so in realistic environments those robots deployed on higher floors tend to be assigned to stair duty, while those on the ground floor mark the emergency exits. This is a solution that the algorithm finds naturally.

C. Navigation and Other Issues

Our navigational algorithm uses a simple graph-based planner, and an implementation of VFH+ [13] for local obstacle avoidance. The former is responsible for generation of way-points, and the latter for traveling between them. The implementation is written in Python and C using the Player robot device server and the Stage simulator. Virtual device abstractions (described in Vaughan et al. [14]) permit effortless use of the VFH+ routines and adaptive particle-based localization [6].

As the robots move through the environment, they continually monitor the network for a signal indicating that a new robot is available to join the team. When this occurs, the current assignment is preempted and a new one (that includes the newcomer) is calculated. A timeout is used to ensure that all robots which were included in an assignment are still available – if one is no longer functional, the assignment is recalculated. These two actions ensure that the positions occupied by beacons always (up to a time-out frequency) reflect the best possible configuration for the number and constraints of the available robots.

The nodes only communicate in order to share their global pose estimates at the beginning of the assignment step. Implicitly, the robots assume that all the others are performing identical assignments and since the assignment algorithm is deterministic, and all the robots arrive at the same deployment agreement. This method of distributing the task allocation process is described in Gerkey and Matarić [7]. No single robot is responsible for assigning tasks to others, their simultaneous execution of the same algorithm results in a consistent allocation.

D. Implementation Details

The algorithm consists of performing four concurrent processes on each robot.

1) The first process runs adaptive Monte Carlo localization using the laser range finder and odometry data made available through the use of an abstract device component in Player [14].
2) The second process is responsible for the navigation and assignment details; it essentially runs as the FSM shown in Figure 4. The following describes each of the states:

   Initial: The state in which the process begins. After a random wait, the robot switches to the communicate state.
   Communicate: While in this state the robot gathers pose estimates from the other robots within communication range. It stays in this state for a minimum amount time, giving all robots sufficient time to obtain necessary pose estimates. Thereafter, it switches to the allocate state.
   Allocate: The robot builds a utility matrix and solves the assignment problem. It then switches to the navigate state. If the calculation is interrupted by another robot's request for a pose estimate (implying the arrival of newcomer) the robot switches back to the communicate state, aborting the current allocation and permitting a new allocation that includes the new robot to be performed.
   Navigate: The robot generates a sequence of way-points to reach the calculated destination. It can remain in this state for a limited amount of time only; if/when the goal is reached within this time, the robot switches to the deployed state. However, if, the goal is not reached, it switches back to the communicate state. If a pose request is received during navigation, it also results in a transition to the communicate state, so that a new allocation will be performed.
   Deployed: Here, the robot physically stays in place, triggering the sound beacon. After a fixed time, it switches to the communicate state. If the global state of the system has not changed, the robot is allocated back to its current location. If another robot requests a pose estimate, then it switches to the communicate state.
3) The third process takes the pose estimates and sends them to clients (other robots) when requested. It com-

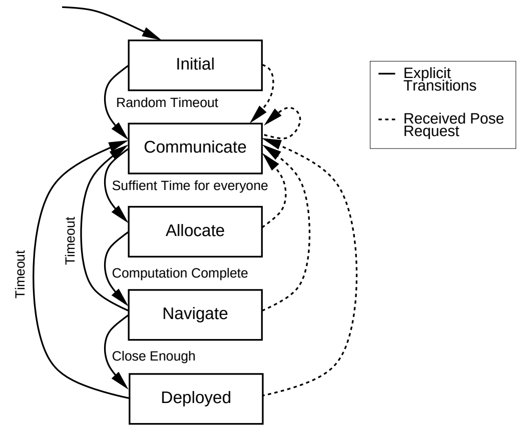

Fig. 4. The Internal State Transition Diagram representation for the robot coordination process. The signal that indicates that a pose request has entered the system is set via another process.

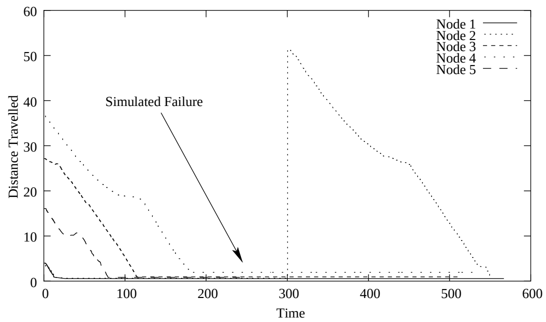

Fig. 5. The distances of each robot from its assigned destination. Note the reassignment occurs at the 300 second mark, resulting in the discontinuity.

municates both with the first process (to get the current pose estimate) and the second (to trigger that someone had requested the current pose information, and that it should switch to the communicate state).

4) The final process permits interactive logging, messaging, and control of certain high-level aspects (shutting down the robot, simulating an error, etc.). This interaction occurs through a shared variable interface in the implementation.

The following table provides a detailed description of the deployment process with five robots and the environment shown in Figure 3.

#	Description
1	The first robot (initially placed on floor 1) partitions the environment into the singleton set $\{\{1, 2, 3, 4, 5, 6, 7\}\}$, and moves toward the closest element; in this case 1.
2	The second robot (initially on floor 1) results in the locations being partitioned into the sets $\{\{1, 2, 3, 4, 6\}, \{5, 7\}\}$, and is assigned the second set; it moves toward 5. The first robot remains at 1.
3	The third robot (initially on floor 1) partitions the locations into the sets $\{\{1, 6\}, \{2, 3, 4\}, \{5, 7\}\}$, and is assigned the second set; it moves to 4. The others remain in their locations (with appropriate assignments).
4	The fourth robot (deployed on floor 2) results in the locations being partition into the sets $\{\{1, 6\}, \{2, 3\}, \{4\}, \{5, 7\}\}$, and is assigned the fourth set; it moves to 7 (since it is on the second floor). The second robot is now forced to move elsewhere, and it moves to 3.
5	The fifth robot (deployed on floor 1) produces $\{\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5, 7\}\}$, and it ends at location 2.

V. Evaluation

We have evaluated our implementation along two dimensions: 1) robustness with respect to failure and 2) effectiveness in terms of impact on the evacuees. A desirable system must meet both of these criteria to a satisfactory degree; we attempt to demonstrate that this is the case for our algorithm.

A. Robustness

Inspired by traditional white-box testing methodologies, we would like to have all possible logic branches of the algorithm tested; this is obviously infeasible. We present a representative sample run where the system behaved reasonably. The run is one of ten instances in which beacons were deployed into the Figure 3 environment and a failure purposefully induced. Similar runs on three additional environments were also performed; in all cases the system demonstrated similar reliability – detection and reassignment to compensate for the failure. The forced failure was performed by shutting down a node once deployed, or during the process of deployment. The exact time of this operation (and its detection) differed from run to run, this fact unfortunately makes it difficult to present a single summary of the data from all the runs in visually meaningful fashion.

Figure 5 shows the distance of each robot from its assigned goal destination for the sample run, over a range of times. (Node 1 and 2 have line segments very close to one another, and are indistinguishable unless magnified.) By time $t = 200$, all of the beacons have been deployed and the system could remain in this state, with activated sound beacons, indefinitely. Next, node 5 was shut down. The system remained in the state, with only four active beacons, until a timeout elapsed resulting in a robot sending a request for another's pose information. This caused a cascade of further requests. The remaining four discovered that node 5 was no longer functioning, and that node 4 would serve as a better beacon if it moved to a place previously assigned to the failed node. This is observable as the spike in the graph around $t = 300$ for node 4, because its destination changed.

The case of a network partition is dealt with similarly; those nodes that can communicate do so, and coordinate their efforts accordingly. Two robots on different partitions cannot communicate, and thus cannot avoid being assigned destinations that are co-located. This, however, has the useful side-effect of bringing them together, and most likely within communication range. One can construct a scenario with communications links being added and removed in a particular order such that a continuous cycle of deployment activities occurs. Our empirical experience indicates that most examples involve some number of robots being stable and repeatedly assigned to the same location, while a few others switching between partitions; those robots that remain stable still act as effective beacons.

An additional ten runs were attempted on three physical robots on the first two floors of our laboratory building (the map is shown in Figure 6). Our experimental platform consisted of three pioneer 2-DX robots equipped with laser range finders. The first two robots were deployed from the second floor; they moved toward the top of the stairs (bottom left corner, Figure 6) and the exit marked 2 (upper right corner, Figure 6) respectively. A third robot was deployed on the bottom floor, it moved toward the exit marked 1. A failure was induced in the robot located at exit 2, after being detected, the robot above the stairwell moved toward exit 2, resulting in a better spread. After six consecutive successful trials, the experiment was halted due to a hardware failure.

B. Effectiveness

There is no formal description for where the nodes should best be deployed, or which beacons should be relocated when a

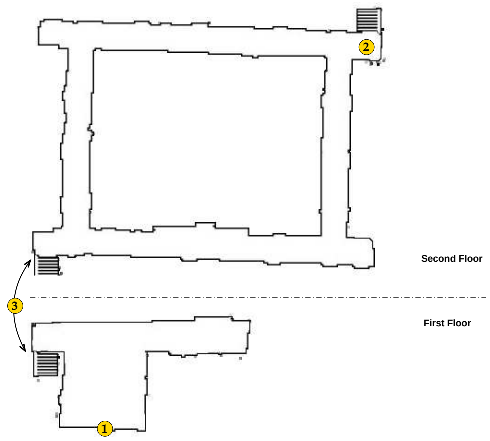

Fig. 6. The map used with physical robots. Exits marked 1, 2. The connection labeled 3 is a stairwell.

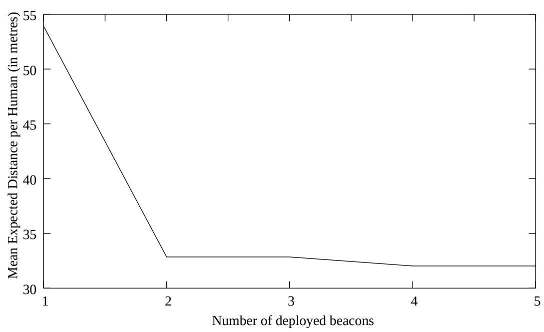

Fig. 7. The mean distance a human will travel (based on the simple flow model described in the text) as a function of the number of deployed beacons. This is a function of the *locations* of the beacons, too; this plot is for the five locations chosen by the described allocation algorithm.

beacon failure occurs. We have embraced the standard practice of using exits and stairways, but evaluation of our location selection mechanism is difficult.

We propose a metric that attempts to capture the efficacy of a given arrangement of beacons. Although extremely simplistic, employing an almost trivial model of people's behavior, it still provides some valuable information.

We define a *choice point* as a location where there is fork in the topological structure of the environment, i.e., where the evacuee (or robot) must make a navigational choice. Suppose that without any audio cues people are likely to pick one of the routes from any given choice point with equal probability. With beacons installed we can consider the choices at each junction as *biased*. In our evaluation, we assume that people are 80% more likely to take a route indicated by the nearest beacon than otherwise. The parameter is a reasonable guess, which we are not able to justify, but which is easily adapted in presence of real-world data.

The model enables us to compute the distance than an evacuee would travel before exiting the building for a particular arrangement of beacons. The distance is measured from all decision points in the environment, and the mean computed.

Figure 7 shows the relationship between expected evacuation distance (or time) and the number of robots deployed into a particular environment. The robots were deployed from the same initial locations as the sample run from the previous section. The environment shown in Figure 3 is used here because it is the most complex of the environments we considered, enabling more robots to be deployed and hence more data points. The trend is, however, consistent with the other simulated deployment runs mentioned in the previous section: a very rapid decrease is effectiveness per beacon. This indicates that even very few beacons are likely to be effective.

In reality, adding too many beacons would saturate the system and begin to have a negative effect. These results do not include those estimations.

VI. Related Work

Using robots in situations involving fire hazards is not new (see for example Kobayashi and Nakamura [11]), however the assistive approach we propose is unique (to the authors' knowledge).

There are a host of other mobile deployment algorithms, a number of them focusing on sensor-based deployment [1, 9, 10]; this is distinct from the functionality we attempt to afford our robots. This other work focuses on distributing sensors into the environment, our work addresses the converse problem: deploying a network capable of no sensing but only actuation. Therefore, rather than focusing on sensor networks, we focus on networks of actuators (robots and sound beacons).

Task allocation is a widely studied problem in the multi-robot and multi-agent communities, and formal frameworks are starting to emerge (see for example the complexity bounds in Gerkey and Matarić [7]). The navigation work presented in Berhault et al. [2] describes a methodology for allocating multi-task robots and single-robot tasks. They make use of bundles of tasks, where a robot commits to performing a number of tasks (or navigating to a set of places). This is related to our work, because they aim to ensure that a robot will be assigned multiple tasks, and we aim to ensure that other robots are not assigned near by locations. In their work spatial clustering mechanisms are used for producing the task bundles, but an auction-based methodology is used to arrive at the final assignment.

Much of our implementation was aided by the use of existing abstract devices [14] in the Player/Stage device server and simulation suite, including particle-based localization [6] and VFH+ for local obstacle avoidance [13], which were used as existing components.

VII. Conclusion and Future Work

This paper presents a justification for a new multi-robot task domain, self-deployment of assistive audio beacons, and a description of a system we implemented in a realistic simulation environment and physical robots that effectively performs this task. We have argued that this assistive method should complement existing robotic (and other) approaches to disaster management and recovery.

While reuse is far from pervasive in robotics, we have leaned heavily on existing work and known solutions to well understood problems. We have also shown that creative approaches to the generation of tasks permits some of the shortcomings of existing task allocation systems to be overcome in a domain-specific manner.

Our future work in this area will consider the possibility of the network nodes containing limited sensing capabilities, so that rough estimates of the number of people involved in evacuation can be estimated. This will allow for the possibility of the network performing "routing" and "load balancing" operations. We will also pursue more detailed evaluation of the effectiveness of self-deployed beacons with human subjects (although not in emergency situations). The experimental work here does not use sophisticated models for predicting evacuation routes of people and could be significantly improved (see for example Hamacher and Tjandra [8]). We are engaged in developing models of crowd behavior for this and related purposes.

Finally, we believe that the algorithm presented herein, if moved to sufficiently robust and capable hardware, could serve to save lives in a real-world evacuation scenario.

ACKNOWLEDGMENTS

This work is supported in part by the DARPA MARS DABT63-99-1-0015 and TASK F30602-00-2-0573 grants, and in part by ONR MURI N00014-01-1-0890 grant.

We thank the Player/Stage developers Vaughan et al. [14], specifically Chris Jones for his implementation of the local obstacle avoiding navigation algorithm and Andrew Howard for his adaptive MCL device. Chad Jenkins' editing input was extensive and extraordinary.

REFERENCES

[1] M. Batalin and G. Sukhatme. Coverage, exploration and deployment by a mobile robot and communication network. In *Proceedings of the International Workshop on Information Processing in Sensor Networks*, Palo Alto Research Center (PARC), Palo Alto, CA, USA, April 2003.

[2] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt. Robot exploration with combinatorial auctions. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS'03)*, Las Vegas, NV, USA, October 2003.

[3] J. Casper, M. Micire, and R. Murphy. Issues in intelligent robots for search and rescue. *SPIE Ground Vehicle Technology II*, 4: 41–46, 2000.

[4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 2nd edition, 2001.

[5] Dusseldorf Airport Disaster, 10 April 1997. Official Prosecutor Investivative Report.

[6] D. Fox. KLD-sampling: Adaptive particle filters. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 713–720, Cambridge, MA, USA, 2001. MIT Press.

[7] B.P. Gerkey and M. Matarić. A formal framework for the study of task allocation in multi-robot systems. Technical Report CRES-03-013, Center for Robotics and Embedded Systems, University of Southern California, 2003.

[8] H.W. Hamacher and S.A. Tjandra. Mathematical modelling of evacuation problems: A state of the art. In *Pedestrian and Evacuatin Dynamics*, pages 227–266, 2001.

[9] A. Howard, M.J. Matarić, and G.S. Sukhatme. An incremental deployment algorithm for mobile robot teams. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2849–2854, EPFL Switzerland, 2002.

[10] A. Howard, M.J. Matarić, and G.S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5: Proceedings of the 6th International Conference on Distributed Autonomous Robotic Systems (DARS02) Proc. Fifteenth Annual ACM*, pages 299–308, Fukuoka Japan, 2002.

[11] A. Kobayashi and K. Nakamura. Rescue robots for fire hazards. In *Proc. of the first International Conference on Advanced Robotics (ICAR'83)*, pages 91–98, September 1983.

[12] S. Thrun and A. Bucken. Integrating grid-based and topological maps for mobile robot navigation. In *AAAI/IAAI, Vol. 2*, pages 944–950, 1996.

[13] I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA'98)*, pages 1572–1577, Leuven, Belgium, May 1998.

[14] R.T. Vaughan, B.P. Gerkey, and A. Howard. On device abstractions for portable, reusable robot code. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2421–2427, Las Vegas, NV, USA, October 2003.

[15] D. Withington. Life saving applications of directional sound. In *Pedestrian and Evacuatin Dynamics*, pages 277–298, 2001.