Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

CoBots: Robust Symbiotic Autonomous Mobile Service Robots

Manuela Veloso, Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal
School of Computer Science
Carnegie Mellon University
mmv@cs.cmu.edu, joydeepb@ri.cmu.edu, brian@coltin.org, steph.rosenthal@gmail.com

Abstract

We research and develop autonomous mobile service robots as Collaborative Robots, i.e., CoBots. For the last three years, our four CoBots have autonomously navigated in our multi-floor office buildings for more than 1,000km, as the result of the integration of multiple perceptual, cognitive, and actuations representations and algorithms. In this paper, we identify a few core aspects of our CoBots underlying their robust functionality. The reliable mobility in the varying indoor environments comes from a novel episodic non-Markov localization. Service tasks requested by users are the input to a scheduler that can consider different types of constraints, including transfers among multiple robots. With symbiotic autonomy, the CoBots proactively seek external sources of help to fill-in for their inevitable occasional limitations. We present sampled results from a deployment and conclude with a brief review of other features of our service robots.

1 Introduction

We research, develop, and deploy multiple autonomous mobile robots capable of performing tasks requested by users in our multi-floor office building. To successfuly perform service tasks, our robots have several core capabilities, namely:

- To autonomously localize and navigate in the diverse types of indoor space, including corridors, elevators, and open areas with movable furniture and people.
- To schedule conflict-free plans for multiple robots to satisfy constrained tasks specified and requested by users.
- To overcome the robots' own limitations, in particular in actuation, by proactively ask for help from humans.

Our current four Collaborative Robots, CoBot-1 through CoBot-4 (see Figure 1) can be viewed as mobile, computing, and sensing platforms, that behave as service robots.[1]

Mobile robots, by definition, need to be able to move, in our case, in indoor environments. Such capability has been extensively investigated. The fact that our research goal includes the persistent deployment of the CoBots led to the introduction of novel mapping, sensing, and localization approaches [Biswas, 2014]. The robots classify sensed obstacles as map-known long-term features (walls) or map-missing short-term (furniture) and dynamic (people) features. This explicit distinction enables the overall effective episodic non-Markovian localization approach [Biswas *et al.*, 2011; Biswas and Veloso, 2012; 2013].

Figure 1: CoBots with omnidirectional motion, onboard computation, interaction interfaces, carrying baskets, and different combinations of depth sensing (cameras and LIDAR).

Service robots need to be able to perform service tasks. The CoBot robots can perform multiple classes of tasks, as requested by users through a website [Ventura *et al.*, 2013], in person through speech [Kollar *et al.*, 2013] or on the robot's touch screen. All tasks can be represented as pick up and delivery tasks of objects or people. The task scheduler takes into account time and location constraints, as well as the multiple available robots, and issues plans that can include transfers [Coltin, 2014].

As can be seen in Figure 1, a CoBot has no hands but has a basket, so it can carry, but not manipulate items. To overcome this actuation limitation, and inevitably other types of limitations, the robots proactively ask for help from humans [Rosenthal *et al.*, 2010; 2011], and from the web [Samadi *et al.*, 2012]. They can gather and use models of human help and preferences in a human-centered planning approach [Rosenthal, 2012].

[1]Thanks to Mike Licitra for designing and building the CoBots, and to Joydeep Biswas for keeping them functional.

2 Episodic non-Markov Localization

A variety of early robots, such as Shakey [Nilsson, 1984]), Xavier [Simmons *et al.*, 1997], and museum tour guide robots [Burgard *et al.*, 1999; Fong *et al.*, 2003], and more recent ones [Chen *et al.*, 2012; Randelli *et al.*, 2013; Christensen *et al.*, 2010; Hawes *et al.*, 2007; Dias and Ventura, 2013; Zhang and Stone, 2015; Visser and Burkhard, 2007]. All of these efforts include variations of localization algorithms [Dellaert *et al.*, 1999]. Our CoBot robots, as deployed in a multi-floor university building setting, now for more than 1,000km, have faced new challenges.

Over the course of their regular deployments, the CoBots are exposed to a variety of types of environments. Some environments like corridors remain largely invariant over time, with little or no changes. Other environments like cafe areas and open atria, exhibit significant changes over time, with objects like tables and chairs being moved around frequently, and numerous dynamic obstacles like humans. Such environments pose a challenge to localization algorithms that assume that the world can be represented by a static map.

To localize in the presence of frequently observed movable and moving objects, we introduce Episodic non-Markov Localization [Biswas and Veloso, 2014] that explicitly reasons about observations of non-mapped objects without saving locally static maps. Episodic non-Markov localization maintains a belief of the history of pose estimates of the robot over "episodes" of observations of unmapped objects. For every time-step, it classifies observations into those arising from the map ("Long Term Features", LTFs), from unmapped static objects ("Short Term Features", STFs), or from moving objects ("Dynamic Features", DFs). The correlations between poses of the robot due to the presence of STFs and DFs are represented by a "Varying Graphical Network" (VGN), which we introduce next.

2.1 The Varying Graphical Network

As in a Dynamic Bayesian Network, a VGN includes certain periodically repeating nodes and edges that do not change with the belief. We term these the non-varying nodes and edges. A VGN includes two additional structural elements: varying nodes and varying edges. The presence and structure of the varying nodes and varying edges are not known a-priori, and are estimated jointly with the belief. Since the estimates of the structure may change with the belief, the structure is likely to change as new observations become available.

VGNs provide an accurate representation for non-Markov localization. The presence of LTFs and their relations to the map, and the correlations between successive poses of the robot due to odometry observations are encoded by the non-varying edges and nodes. The presence of STFs and DFs is encoded by the presence of associated varying nodes. The correlations between STFs observed at different time-steps is encoded by the varying edges. The Belief of the robot's localization, $Bel(x_{1:n})$ is maintained over a history of n poses $x_{1:n}$. For each timestep i, odometry u_i corresponds to the robot's relative motion between poses x_{i-1} and x_i, and observation s_i, made at pose x_i, includes observations of LTFs that match the map, as well as unexpected observations of STFs and DFs.

Since the VGN for non-Markov localization has no predefined structure, it might seem that the computation of the belief would require storing the complete history of all states and observations since the robot was turned on. However, in practice this is not necessary, as we rely on the existence of "episodes" in non-Markov localization. Suppose there exists a time step t_i such that all observations and state estimates made after t_i, given x_i, are independent of all prior observations and state estimates:

$$P(x_{1:n}|x_0, s_{1:n}, u_{1:n}, M) = \\ P(x_{1:i}|x_0, s_{1:i}, u_{1:i}, M) \times P(x_{i+1:n}|x_i, s_{i+1:n}, u_{i+1:n}, M). \quad (1)$$

This conditional independence implies that there are no STF observations after t_i that correspond to STF observations before t_i. In such a case, the history of states and observations prior to t_i, called the "episode" $t_{0:i-1}$, can be discarded when estimating $Bel(x_{i:n})$ over the episode $t_{i:n}$. We assume such episode-boundary time-steps like t_i exist, allowing real-time non-Markov localization with limited computational resources. Figure 2 shows an example VGN near an episode boundary, highlighting the absence of any varying edges crossing the episode boundary.

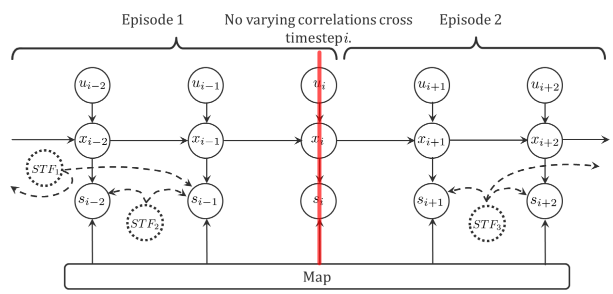


Figure 2: An example VGN demonstrating the presence of an episode in non-Markov localization. Note the absence of any varying edges that cross the red line indicating the episode boundary. Hence the pose x_i is an episode boundary, where all previous poses up to x_{i-1} are in the previous episode, and poses x_i and later are in the latest episode. Observations s_{i-1} and older thus no longer need to be stored.

The exact structure of the VGN depends on the specific LTFs and STFs that are observed by the robot, and we next present how the observations are classified.

2.2 Classification of Long-Term and Short-Term Features

For every time-step, the structure of the VGN, based on the classification of the observations into LTFs, STFs and DFs, is re-evaluated prior to updating the MLE of the belief. In this work the sensor we use is a laser rangefinder, so each observation s_i is a set of n_i 2D points $s_i = \{p^i_j\}_{j=1:n_i}$ observed by the robot. We represent the pose x_i of the robot on the map at time-step i as an affine transform T_i that consists of a 2D rotation followed by a 2D translation.

We use a vector map [Biswas *et al.*, 2011] representation $M_{\text{vector}} = \{l_i\}_{1=1:s}$ for the permanent map, consisting of a set of s line segments l_i. To evaluate which of the observed points p_k^i are LTFs, an analytic ray cast [Biswas and Veloso, 2012] is performed from the latest MLE of x_i. The result of the analytic ray cast is a mapping from $p_j^i \rightarrow l_j \in M_{\text{vector}}$, indicating that the line segment l_j from map M_{vector} is the most likely line in the map to be observed by the point p_j^i. Let d$ist(p, l)$ denote the perpendicular distance of point p from the line segment l where both p and l are in the reference frame of the map. The observation likelihood $P(p_j^i|T_i, M_{\text{vector}})$ of the point p_j^i is then given by

$$P(p_j^i|x_i, M_{\text{vector}}) = \exp\left(-\frac{\text{d}ist(T_i p_j^i, l_j)^2}{\Sigma_s}\right), \quad (2)$$

where Σ_s is the scalar variance of observations, which depends on the accuracy of the sensor used. Thus, observations are classified as LTFs if the observation likelihood of the point given the map is greater than a threshold, $P(p_j^i|x_i, M_{\text{vector}}) > \epsilon_{\text{LTF}}$.

Observed points that are classified as non-LTFs could potentially be STFs. To check if an observed point $p_j^i \in \overline{\text{LTF}}_i$ is an STF, it is compared to all non-LTF points observed prior to time-step i to check if they correspond to observations of the same point. Given a point $p_j^i \in \overline{\text{LTF}}_i$ observed at time-step i and another point $p_k^l \in \overline{\text{LTF}}_l$ observed at a previous time-step l, the probability that both the observations correspond to the same point is given by the STF observation likelihood function,

$$P(p_j^i, p_k^l|x_i, x_l) = \exp\left(-\frac{||T_i p_j^i - T_l p_k^l||^2}{\Sigma_s}\right). \quad (3)$$

Therefore, a non-LTF point $p_j^i \in \overline{\text{LTF}}_i$ is classified as an STF if there exists a point $p_k^l \in \overline{\text{LTF}}_l$ from a time-step $l, l < i$ such that $P(p_j^i, p_k^l|x_i, x_l) > \epsilon_{\text{STF}}$.

Given the classifications, and the form of the observation likelihoods of the LTFs and STFs, episodic non-Markov localization solves for the maximum likelihood estimate of the belief by representing the Belief as a cost function and optimizing over it, instead of keeping multiple estimates represented as a particle filter [Biswas, 2014].

We convert the belief from a probability distribution representation to a cost function representation C such that

$$\begin{aligned} Bel(x_{1:n}) &= P(x_{1:n}|x_0, s_{1:n}, u_{1:n}, M) \\ &\propto \exp(-C(x_{1:n}|x_0, s_{1:n}, u_{1:n}, M)). \end{aligned} \quad (4)$$

The cost function C consists of a sum of m sub-cost functions c_j^{STF} corresponding to the STF terms $P(s_{1:n}^{\text{STF}_j}|x_{1:n})$, n sub-cost functions c_i^{LTF} corresponding to the LTF terms $P(s_i^{\text{LTF}}|x_i, M)$, and n sub-cost functions c_i^{odom} corresponding to the odometry terms $P(x_i|x_{i-1}, u_i)$.

The Maximum Likelihood Estimate $x_{1:n}^*$ is therefore computed by minimizing the cost function as:

$$x_{1:n}^* = \arg\min_{x_{1:n}} \left(C(x_{1:n}|x_0, s_{1:n}, u_{1:n}, M)\right). \quad (5)$$

Thus, Episodic non-Markov Localization updates the maximum likelihood location estimates of the robot via functional non-linear least squares optimization of Equation 5.

2.3 Results

Episodic non-Markov Localization has been deployed on all the CoBots over part of a 1,000km Challenge [Biswas, 2014], and has been used to localize the robots in many different environments spanning multiple floors across multiple buildings. In particular, it has been instrumental in increasing the robustness of localization on floors with challenging open areas, like a large atrium on the floor GHC4. Figure 3 illustrates different placements of the STFs, namely movable furniture.

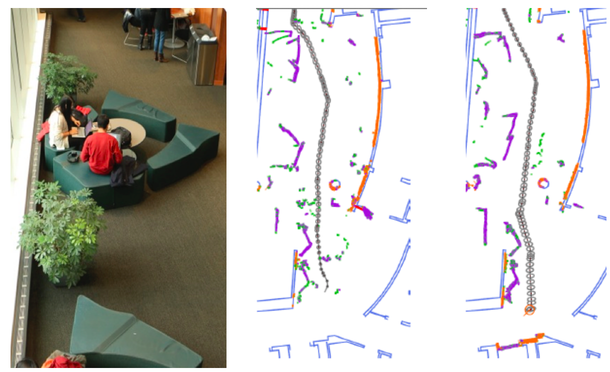


Figure 3: View of the challenging varying space in GHC4 atrium and snapshots of enML at two different times. The trajectory of the robot over the episode is shown in grey, along with the covariance elipses. LTF observations are shown as orange points, STF observations as purple points, and DF observations as green points. The long-term static map is shown as blue lines.

To highlight the contribution of the robustness of Episodic non-Markov localization to the deployments of the CoBots, we tabulate the mean distance traversed autonomously by the CoBots between operator interventions in Table 1.

	CGR	EnML
GHC4	0.62	4.42
GHC6	8.61	9.48
GHC7	5.58	9.02
GHC8	6.04	19.36
GHC9	5.33	20.05
NSH4	0.56	2.65
All	4.79	8.13

Table 1: Mean distance (in km) traversed between interventions using CGR (a variant of Markov Localization) and EnML per map over the 1,000km Challenge.

The CoBots traversed a mean distance of $4.42km$ between interventions while using EnML for localization as opposed to $0.62km$ when using Corrective Gradient Refinement (CGR) [Biswas *et al.*, 2011], a variant of Markov Localization. Overall, EnML allowed the CoBots to traverse a mean of $8.13km$ as opposed to $4.79km$ when using CGR. The increased robusteness is attributed to the ability of EnML to reason better about observations of unmapped objects, and hence its robustness to changes in the environment.

3 Scheduling for Transfers with CoBots

Tasks requested by users are processed by a *scheduler* that computes an ordered assignment of tasks to the multiple robots. The scheduler needs to satisfy various constraints stated by the users, including location, time windows, transportation capacities of the robots, and maximum delivery times. The goal of scheduler is to find a valid schedule which minimize the total distance traveled by the robots and, or the completion times of the tasks. The scheduler outputs task execution times for each robot, and sends lists of tasks to the robots. During execution, the robots update the scheduler of their progress.

We realized that the robots can perform their tasks more efficiently by *transferring* items between one another [Coltin, 2014]. For example, the scheduler, without considering transfers, could assign CoBot-1 and CoBot-2 both to pick up items on the seventh floor that they need to deliver to the ninth floor. Instead of both taking an elevator ride, CoBot-1 could transfer its item to CoBot-2, which could deliver both items.

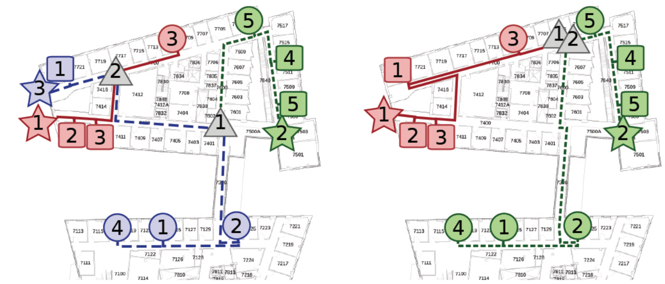
Initially, we introduce a scheduler that generates an optimal schedule for the CoBots using mixed integer programming (MIP) [Coltin, 2014]. Finding the optimal schedule is NP-hard, so the MIP solver scales poorly, although for our typical usage of less than fifteen tasks at once solving the problem optimally is feasible.

To scale to larger problems, we developed an approximation algorithm for a variant of the scheduling problem in which all the items share the same destination and there are no time constraints. This is a common scenario for the CoBots, when they pick up mail for delivery to the central office, or hand out candy to building occupants for Halloween. The approximation algorithm is based on an approximation for the traveling salesman problem, and returns a solution that is guaranteed to be within a factor of two of optimal in terms of total distance traveled [Coltin and Veloso, 2014a].

Expanding to the more general problem in which items have distinct destinations, we introduced three heuristics to from schedules, still without considering time: a greedy approach, an algorithm based on auctions, and an algorithm where an item's entire trajectory is inserted into a graph of transfers. The heuristics reduce the search space by inserting transfers into existing schedules, and hence may not find the optimal solution. Transfers were shown to reduce the solution cost compared to similar heuristics without transfers [Coltin and Veloso, 2014b].

We extended the auction heuristic to work with time windows, by determining execution times through the use of simple temporal networks. The auction algorithm is applied online as new tasks come in from users, so that the CoBots replan online. If a CoBot is delayed or disabled, the other CoBots replan so that the tasks are still completed as quickly as possible (see Figure 4).

The CoBots also take advantage of the fact that there are multiple robots to replan better schedules. If a robot is blocked in a hallway, it will inform the other robots it is blocked. The other robots will then replan to avoid the blocked hallway, if possible, as shown in Figure 5. Additionally, robots detect if doors are open or closed when they drive past. If one robot happens to drive by a closed door that another robot is planning to pick up or deliver an item from, it will tell the other robot, and the scheduler will attempt to delay the task at that room until a later time when the occupant has hopefully returned to their office [Coltin and Veloso, 2013]. In general, each robot can be aware of and check the *rationale* of the plans of other robots.



(a) (b)

Figure 4: (a) Deliveries are scheduled with three robots, including two transfers; (b) When one of the robots fails, the tasks are rescheduled. Squares indicate pickups, circles indicate deliveries, triangles indicate transfers, and stars indicate robot starting points. The numbers inside indicate either robot and item numbers.

Finally, we introduce an algorithm based on simulated annealing which finds high quality non-optimal schedules with transfers. While more computationally expensive than the previous heuristics, this algorithm outperforms the best previous solutions to benchmark scheduling problems by incorporating transfers. In addition to the CoBots, the idea of transfers are also applied to transportation and ridesharing problems to reduce fuel costs [Coltin and Veloso, 2014c].

Multiple CoBots continue to autonomously perform tasks in the Gates-Hillman building. The scheduling algorithms, with transfers, allow the CoBots to complete more tasks, more quickly, while prolonging battery life.

4 Human-Centered Planning for Symbiotic Autonomy

Rather than limiting robots' tasks to those that only include actions that robots can perform autonomously, CoBot instead reasons about, plans for, and overcomes its limitations by proactively asking humans in the environment for help [Rosenthal *et al.*, 2010].

We introduced a human-centered planning algorithm that asks for help when CoBot is uncertain of its location or when it is uncertain of which action to take [Rosenthal *et al.*, 2010]. Robots and humans are in a symbiotic relationship, as robots perform service tasks for humans, and humans may need to help the robots. The underlying assumption for the symbiotic robot autonomy is that the requests for help from the robot, e.g., pressing an elevator button, are simple for humans.

The symbiotic autonomy approach leads to adding *ask-for-help* action primitives to the robots' plans. The robots autonomously perform such actions. Figure 6 shows a high-level partial conditional plan for the robot to navigate to a

(a) CoBot-2 to Office (b) CoBot-4 Blocked (c) CoBot-2 Replans (d) Obstacle Avoided

Figure 5: (a) CoBot-2 heads towards an office to make a delivery, and shares with CoBot-4 the path that it needs to traverse; (b) CoBot-4 detects that a hallway of relevance to CoBot-2's path is blocked; (c) The scheduler replans for CoBot-2; and (d) CoBot-2 takes an alternate round to avoid the blocked hallway.

room, where it asks for help from a human to push the elevator buttons.

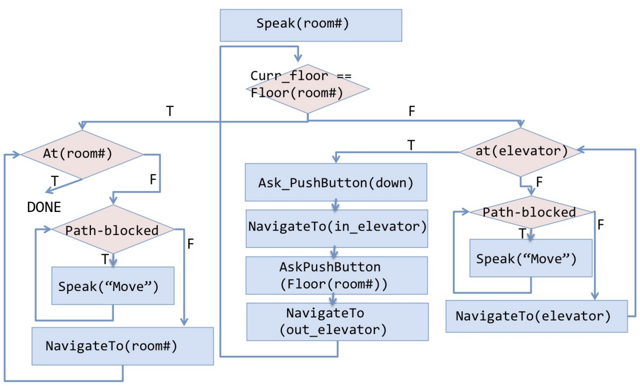


Figure 6: High-level partial conditional plan for symbiotic autonomy to navigate with actuation limitations and asking for help if needing to take the elevator.

The conditional plan in Figure 7 is partial in the sense that it does not include what happens if any of the actions fails. We have developed several approaches to handle the additional contingencies of symbiotic autonomy, namely (i) no human helps, e.g., the robot keeps waiting by the elevator; (ii) a human provides the wrong help, e.g., it tells the robot is on an incorrect floor. In the latter situation, as soon as the robot recognizes that it is not at its desired location, it continues its execution by replanning or recognizing that it cannot perform its task for any reason. In both situations, the robot is unable to proceed its task. We have developed two approaches: timeout-based one and a proactive-seek for help one. In the timeout-based approach, the robot waits for help for a predefined amount of time, after which it sends email to its developers using a template that it fills in describing the location and situation where it finds stalled. This step represents an action to ask for help from remote humans.

In the proactive-seek for help approach, we studied who, whether, and where to *proactively* ask building occupants for help, concretely to use the elevator. We made five hypotheses based on our intuitions about what human state attributes matter in determining where and who to ask for help. The first two hypotheses represent the spatial considerations that CoBot robot should take into account.

- Cost of Help: Asking someone for help who is already at the elevator is preferred over finding someone in an office. A benefit of asking the elevator person is that they are already performing the action themselves and should have little cost to helping the robot
- Distance to Help Location: If someone in an office must be asked because it is unlikely that anyone will be at the help location, there should be a preference for asking someone close to the location to avoid making someone travel too far. Although CoBot is mobile and are capable of traveling to find help, an in-office helper would have to travel back to the help location.

The second three hypotheses represent the considerations the robot should make to increase the likelihood that people are willing to comply and help the robot, because the robot need help performing these actions over a long period of time.

- Interruption: The robot should avoid requesting help from people in offices that are likely to be busy.
- Recency of Last Question and Frequency of Questions: The robot should take into account how recently it asked different helpers to avoid asking too often.
- Availability: If a robot travels with a person to the help location and there is someone already at the location of help, the traveling person may feel that they were asked unnecessarily.

Through user studies, we confirmed all five hypotheses [Rosenthal and Veloso, 2012]. Robots should consider the cost of help, distance to help location, availability, interruptibility, and frequency and recency of questions. However,

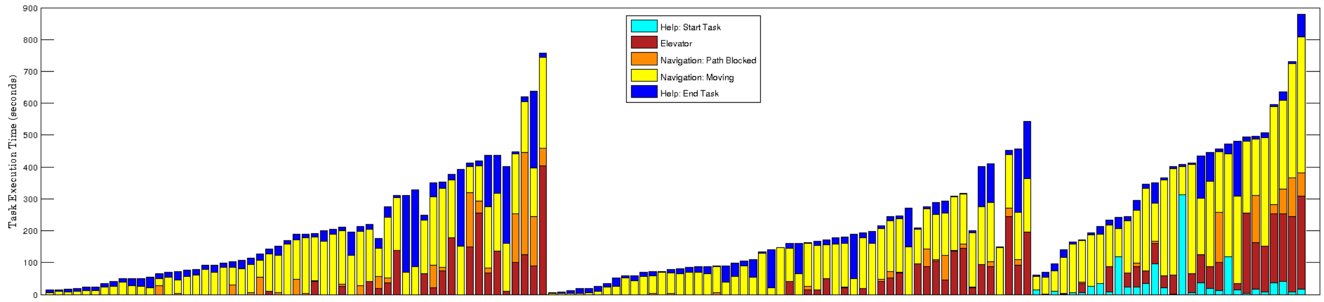


Figure 7: Execution times for, from left to right, Deliver Message tasks, Go to Room tasks, and Transport tasks. The breakdown includes 1) waiting for help to start the task, 2) riding the elevator, 3) navigating (not including time blocked by obstacles), 4) waiting blocked by an obstacle, and 5) waiting for help to end the task.

some participants were willing to help irrespective of the distance to the help location. We use these human state attributes in our human-centered planning algorithm to determine who to ask for help and where to navigate.

When CoBot needs help to use the elevator, it first asks at the elevator hall if anyone is available to help. The people at the elevator hall location have the lowest cost of helping the robot because they are at the elevator anyway. However, if no one helps CoBot, it plans where to seek for help by computing the decision-theoretic expected cost of asking a person in their office based on our user study findings to come to help the robot to get to the desired floor.

The goal of our human-centered proactive replanning algorithm is to simultaneously reduce the time to complete the task while also limiting the in-office help [Rosenthal and Veloso, 2012]. We were able to show that CoBot could complete tasks 4mn faster on average with the proactive replanning algorithm compared to waiting at the elevator only.

The CoBot robots have been deployed for more than 1,000km, in our multi-floor buildings successfully navigating, using their symbiotic autonomy, in particular to move between floors. We present a sample of the results to illustrate the impact of the symbiotic autonomy in the timing of the tasks. The results correspond to a deployment of one robot on the upper four floors of our office building for a two week period. CoBot was deployed for two hours every weekday and made available to the building occupants.

The response to CoBot's deployment was positive: over 100 building occupants registered to use CoBot. Users found creative ways to exploit the robot's capabilities, including, but not limited to sending messages to friends, reminding occupants of meetings, escorting visitors between offices, delivering printouts, inter-office mail, USB sticks, snacks, owed money, and beverages to other building occupants.

We found that occupants scheduled the robot to transport objects between multiple floors of the building more often than they used the multi-floor functionality for other tasks (see Table 2). In particular, the transport task saved the task solicitors time because they did not have to travel between floors themselves. However, even the other scheduled tasks utilized the elevator 40% of the time.

Figure 7 shows how much time CoBot took to execute each task, and how that time was apportioned. A total of

Table 2: Total number of task requests per task type and the respective number that used the elevator.

Task Type	Total Requests	# Multi-floor
Escort	3	2
GoToRoom	52	22
DeliverMessage	56	20
Transport	29	22

140 tasks were completed during the two-week deployment, which took 9 hours and 13 minutes. Based on these times, we find that task solicitors quickly responded to the robot's request for help at the start and end of tasks. Building occupants (even those that had never scheduled a task) were willing and able to help the robot in and out of the elevator. This finding supports our model of symbiotic autonomy, namely that humans are willing to help a robot complete its tasks so that the robot is available and capable of performing tasks for them as well at another time.

5 Conclusion

The CoBot robots have been successfully deployed in multi-floor buildings for over three years. We summarized some of the core contributions. The episodic non-Markovian localization to effectively handle environments whose depth appearance varies over time. Long-term features, e.g., walls, match existing floor-plan maps, while short-term features, e.g., furniture, match previous observations in an episodic non-Markovian manner. The multi-robot task scheduler considers transfers among robots to optimize the travel time performance, and replans to handle online requests and changing conditions. Symbiotic autonomy enables the robot to ask for help from humans at the place needed or proactively search for help from near-by humans. Human-centered planning uses models of humans to generate robots' plans.

Current and future work include learning to improve service performance, including human-preference and environment learning and exploration. We also continue to research on detection of anomalies for safety of use. We are also focused on task instruction and correction through natural language [Mericli *et al.*, 2014], to enable any user to requests new tasks from the robot.

Acknowledgements

We thank Aijun Bai, Steven Klee, Thomas Kollar, Max Korein, Juan Pablo Mendoza, Cetin Mericli, Daniele Nardi, Vittorio Perera, Mehdi Samadi, Yichao Sun, and Rodrigo Ventura for their contributions to the CoBot robots.

This research is partly sponsored by the NSF under awards NSF IIS-1012733 and IIS-1218932, and the ONR under grant number N00014-09-1-1031. The views and conclusions contained herein are those of the authors only.

References

[Biswas and Veloso, 2012] J. Biswas and M. Veloso. Depth camera based indoor mobile robot localization and navigation. In *Proc. of ICRA*, pages 1697–1702, 2012.

[Biswas and Veloso, 2013] J. Biswas and M. Veloso. Localization of the CoBots over Long-term Deployments. *Int. Journal of Robotics Research*, 32(14):1679–1694, 2013.

[Biswas and Veloso, 2014] J. Biswas and M. Veloso. Episodic non-markov localization: Reasoning about short-term and long-term features. In *Proc. ICRA*, 2014.

[Biswas *et al.*, 2011] J. Biswas, B. Coltin, and M. Veloso. Corrective gradient refinement for mobile robot localization. In *Proceedings of IROS*, pages 73–78, 2011.

[Biswas, 2014] Joydeep Biswas. *Vector Map-Based, Non-Markov Localization for Long-Term Deployment of Autonomous Mobile Robot*. PhD thesis, Carnegie Mellon University, 2014. CMU-RI-14-25.

[Burgard *et al.*, 1999] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114:3–55, 1999.

[Chen *et al.*, 2012] X. Chen, J. Xie, J. Ji, and Z. Sui. Toward open knowledge enabling for human-robot interaction. *J. of Human-Robot Interaction*, 1-2:100–117, 2012.

[Christensen *et al.*, 2010] H. I. Christensen, G.J. Kruijff, and J. Wyatt, editors. *Cognitive Systems*. COSMOS. Springer Verlag, Berlin, DE, May 2010.

[Coltin and Veloso, 2013] B. Coltin and M. Veloso. Towards replanning for mobile service robots with shared information. In *Proc. of ARMS Workshop, AAMAS*, 2013.

[Coltin and Veloso, 2014a] B. Coltin and M. Veloso. Optimizing for transfers in a multi-vehicle collection and delivery problem. In *Proc. of DARS*, 2014.

[Coltin and Veloso, 2014b] B. Coltin and M. Veloso. Ridesharing with passenger transfers. In *IROS*, 2014.

[Coltin and Veloso, 2014c] B. Coltin and M. Veloso. Scheduling for transfers in pickup and delivery problems with very large neighborhood search. In *AAAI*, 2014.

[Coltin, 2014] Brian Coltin. *Multi-agent Pickup and Delivery Planning with Transfers*. PhD thesis, Carnegie Mellon University, 2014. CMU-RI-14-05.

[Dellaert *et al.*, 1999] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of ICRA*, 1999.

[Dias and Ventura, 2013] D. Dias and R. Ventura. Absolute localization for low capability robots in structured environments using barcode landmarks. *J. of Automation, Mobile Robotics and Intelligent Systems*, 7(1):28–34, 2013.

[Fong *et al.*, 2003] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42:143–166, 2003.

[Hawes *et al.*, 2007] N. Hawes, A. Sloman, J. Wyatt, M. Zillich, H. Jacobsson, G.-J. Kruijff, M. Brenner, G. Berginc, and D. Skocaj. Towards an Integrated Robot with Multiple Cognitive Functions. In *Proceedings of AAAI*, 2007.

[Kollar *et al.*, 2013] T. Kollar, V. Perera, D. Nardi, and M. Veloso. Learning Environmental Knowledge from Task-Based Human-Robot Dialog. In *Proc. of ICRA*, 2013.

[Mericli *et al.*, 2014] C. Mericli, S. Klee, J. Paparian, and M. Veloso. An Interactive Approach for Situated Task Specification through Verbal Instructions. In *Proc. of AAMAS'*, 2014.

[Nilsson, 1984] N. J. Nilsson. Shakey the robot. Technical Report 323, SRI International, 1984.

[Randelli *et al.*, 2013] G. Randelli, T. Bonanni, L. Iocchi, and D. Nardi. Knowledge acquisition through humanrobot multimodal interaction. *Intelligent Service Robotics*, 6:19–31, 2013.

[Rosenthal and Veloso, 2012] S. Rosenthal and M. Veloso. Mobile Robot Planning to Seek Help with Spatially-Situated Tasks. In *Proc. of AAAI*, 2012.

[Rosenthal *et al.*, 2010] S. Rosenthal, J. Biswas, and M. Veloso. An Effective Personal Mobile Robot Agent Through Symbiotic Human-Robot Interaction. In *Proc. of AAMAS*, 2010.

[Rosenthal *et al.*, 2011] S. Rosenthal, M. Veloso, and A. Dey. Learning Accuracy and Availability of Humans who Help Mobile Robots. In *Proc. of AAAI*, 2011.

[Rosenthal, 2012] Stephanie Rosenthal. *Human-Centered Planning for Effective Task Autonomy*. PhD thesis, Carnegie Mellon University, 2012. CMU-CS-12-110.

[Samadi *et al.*, 2012] Mehdi Samadi, Thomas Kollar, and M. Veloso. Using the Web to Interactively Learn to Find Objects. In *Proc. of AAAI*, 2012.

[Simmons *et al.*, 1997] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan. A layered architecture for office delivery robots. In *Proc. of AGENTS*, 1997.

[Ventura *et al.*, 2013] R. Ventura, B. Coltin, and M. Veloso. Web-Based Remote Assistance to Overcome Robot Perceptual Limitations. In *AAAI Workshop on "Intelligent Robotic Systems"*, 2013.

[Visser and Burkhard, 2007] U. Visser and H.-D. Burkhard. RoboCup: 10 years of achievements and future challenges. *AI Magazine*, 28(2):115–132, Summer 2007.

[Zhang and Stone, 2015] S. Zhang and P. Stone. CORPP: Commonsense Reasoning and Probabilistic Planning, as Applied to Dialog with a Mobile Robot. In *Proc. of AAAI*, 2015.