Autonomous Question Answering with Mobile Robots in Human-Populated Environments

Michael Jae-Yoon Chung, Andrzej Pronobis, Maya Cakmak, Dieter Fox, Rajesh P.N. Rao

***Abstract*— Autonomous mobile robots will soon become ubiquitous in human-populated environments. Besides their typical applications in fetching, delivery, or escorting, such robots present the opportunity to be used as information gathering agents. In this paper, we present an end-to-end framework for enabling a mobile robot to answer questions asked by its users about the robot's environment. Our method takes a natural language question about the state of the environment as input and parses it into an information type and locality. It estimates the optimal viewpoint for capturing an image that contains the requested information, navigates to the viewpoint and dynamically adapts to changes in the environment, and returns the captured image as its response. The same method is also used for answering questions retrospectively, by retrieving information from previously recorded sensory streams. We evaluate our approach with a custom mobile robot deployed in a university building, with questions collected from occupants of the building. We demonstrate our system's ability to respond to these questions in different environmental conditions.**

I. Introduction

Many day-to-day tasks in large human environments, such as office buildings, hospitals, or warehouses, require up-to-date knowledge about the environment. However, such environments are inherently dynamic due to human activity. As a result, a large component of common tasks performed by humans in these environments is to simply collect up-to-date information. Our previous work demonstrated the potential of using mobile robots as information gathering agents for humans through user surveys and a Wizard-of-Oz deployment [1]. In this paper, we present an end-to-end framework for using mobile robots to autonomously answer questions asked by users in a dynamic human-populated environment.

Our framework takes a natural language question (e.g. "Is Mike Chung in the robotics lab?"), parses it into a tuple characterizing the requested information, and computes the optimal robot configuration for capturing the information. To compute the optimal robot configuration, our framework scores candidate robot configurations in terms of visibility of requested information according to a semantically annotated map. We iteratively refine our estimate as more up-to-date information becomes available during the execution of the tasks. This allows us to compute a view maximizing the amount of information captured despite potential dynamic changes such as unexpected occlusions or obstacles that prevent the robot from reaching its initial goal.

The authors are with Computer Science & Engineering, University of Washington, Seattle, Washington 98195-2350, USA {mjyc,pronobis,mcakmak,fox,rao}@cs.washington.edu

We present an implementation of the proposed approach on a custom mobile robot deployed in a university building. We first evaluate our language parser with the natural language questions collected in our previous work [1]. We then evaluate our viewpoint estimation with a subset of questions, asked when the environment is in different states. Finally, we show that the same framework can be used to answer questions retrospectively ("Was Mike Chung in the robotics lab?") by choosing the best frame for answering a question in previously recorded data.

II. Related Work

Previous work has explored many applications of autonomous mobile robots in human environments. These include fetching objects [2]–[5], giving tours as a guide [6]–[8], escorting people to a target location [2], [9], or acting as a kiosk to provide information such as directions [10]–[12]. However, the use of such robots to answer people's questions about the environment has been largely unexplored.

Most closely related to our work is research focused on *object search* in human environments [13]–[18]. These approaches utilize domain knowledge for modeling human environments and reasoning about potential target object locations. Some of them acquire parts of domain knowledge from the web [14], [18], or gather the knowledge required for each object search from the web on-demand [15]. While we consider search as a type of information gathering, our work focuses on a different type of information gathering task that involves checking the state of a specified target. Another line of relevant work focuses on understanding human language in the context of tasks in human environments, such as following natural language directions [19], [20] or spatial modeling of linguistic prepositions [21], [22].

Outside the realm of human-populated environments, the use of robots for information gathering is not a new idea. Mobile robots have been used for exploring and gathering information in challenging environments such as space, underwater, or disaster zones [23]–[26]. Some of these [27]–[30] develop general algorithms for information gathering that could also be applied to human-populated environments; however, this potential has not yet been explored.

III. Formative Study: Wizard-of-Oz Deployment

Our previous work explored the potential of using mobile robots for answering questions about the environment from an end-user perspective [1]. We performed a survey in two university buildings and deployed our robot (Fig. 1) in one of the buildings (Computer Science & Engineering, University

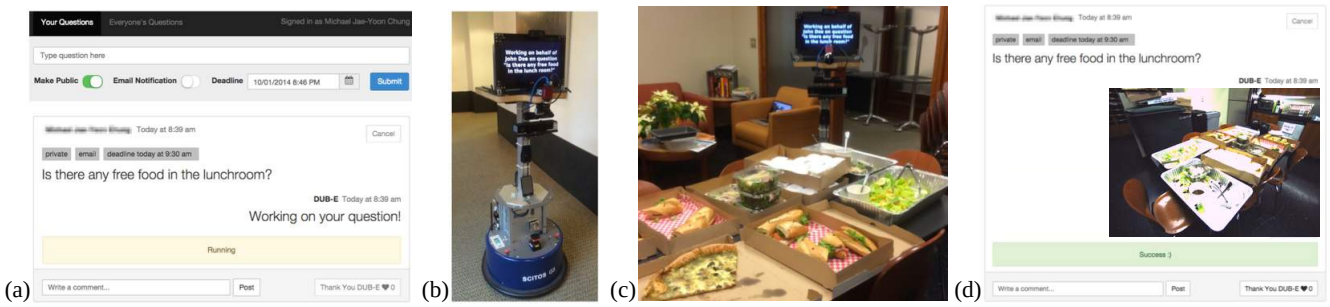

Fig. 1. The end-to-end InfoBot system: (a) a question is submitted through the web interface; (b) InfoBot estimates the initial viewpoint, navigates to the destination while iteratively refining the estimate; (c) it captures the image containing the requested information; (d) the image is delivered to the user through the web interface.

of Washington) using the Wizard-of-Oz technique. As a front-end we created a web interface through which users could post free-form questions and monitor the status of the answer. Users were recruited from graduate and undergraduate students, staff and faculty inhabiting the building.

The experiment was conducted for four business days (9am-5pm). During the experiment, the robot was supervised by a human operator. When a user asked a question from the web interface, the operator received it and accepted it only if it was a valid checking type question that could be answered through a static picture taken by the robot. The operator then supervised the robot to go to the target location in the building, positioned the on-board camera to take a picture. The question was answered by delivering the picture to the web interface.

A. Findings

Over the deployment period, we received 88 valid questions posted by 45 unique users. The majority of questions (71%) were concerned with the **presence** of things at certain locations in the building. Users were mostly interested in the presence of people (33%). Common examples of this type of question are "Is there anyone at {location}?" and "Is {person} in his/her office?" Among questions concerning objects in the environment, users were most interested in the presence of food and mail; e.g. "Is there anything in my mailbox?" and "Is there any food in the downstairs kitchen?"

Another major group of questions were about the **state** of the environment at target locations. Questions ranged from checks about the accessibility of various services (e.g. "Is the door to the conference room open?", "Is the reception still open?") to ambient conditions (e.g. "How noisy is it in the atrium right now?" or "Is it raining outside?").

Overall, 73% of participants used the service more than once. 40% of the users asked more than one question with at least an hour between two consecutive questions. The results from this formative study indicated potential usefulness of the InfoBot, and gave insights into the types of questions people might ask if the robot were to be deployed long-term.

IV. Overview

In this paper we develop an end-to-end framework to autonomously respond to questions asked by users. It is motivated by the types of questions observed in our formative study (Sec. III). We emphasize that our approach aims to capture a sensory snapshot that contains the answer to the user's question, rather than inferring a verbal answer.

A. Problem Description

The goal is to find the best viewpoint v^* in which the requested information $\mathcal{I} \in \{0, 1\}$ is present, given a natural language question s. We assume the robot is operating in a dynamic environment described by a map M, and we have access to the database D containing domain knowledge about M. We formulate our problem as

$$v^* = \operatorname*{argmax}_v P(\mathcal{I} = 1|v, s; M, D) \tag{1}$$

where

$$P(\mathcal{I} = 1|v, s; M, D) = \sum_z P(\mathcal{I} = 1|v, z; M, D)P(z|s; D) \tag{2}$$

$$\approx \max_z P(\mathcal{I} = 1|v, z; M, D)P(z|s; D) \tag{3}$$

where z is a descriptor of the requested information. Factoring the problem in this way allows us to independently estimate the optimal **viewpoint** given a concrete descriptor with $P(\mathcal{I} = 1|v, z; M, D)$ and the **natural language parse** of the question as a descriptor with $P(z|s; D)$.

We define $P(z|s; D)$ as a distribution over information descriptors z for each question sentence s. For example, given the sentence s = "Is there anyone in the robotics lab?" and the record in D that identifies "Mike Chung" as $person$ and "robotics lab" as $cse101$, the desired information descriptor $z = \texttt{presence}(person, cse101)$ is a tuple precisely describing the requested information in s.

The distribution $P(\mathcal{I} = 1|v, z; M, D)$ for estimating viewpoints can be decomposed as follows (M, D omitted to keep notation uncluttered):

$$P(\mathcal{I} = 1|v, z) = \sum_x P(\mathcal{I} = 1|x, z)P(x|v) \tag{4}$$

where x are locations on the map (e.g. cells in a 3D occupancy map), $P(\mathcal{I} = 1|x, z)$ models the presence of the information at location x and $P(x|v)$ models the visibility of location x from viewpoint v.

V. Parsing Natural Language Questions

The deployment experiment in Sec. III revealed that there are two types of checking questions: (a) the questions concerned with *presence* of things at a location, and (b) the questions concerned with *state* of a location. Reflecting this observation, the information descriptor z takes one of two forms: $\texttt{presence}(l, t)$ or $\texttt{state}(l)$, where l is a room in the building (e.g. $cse100$), and t is a target type (e.g. $person$). More formally, we define the information descriptor as a tuple $z = (\tau, l, t)$ where $\tau \in \{\texttt{presence}, \texttt{state}\}$, $l \in \{cse100, cse101, ...\}$, and $t \in \{person, object, N/A\}$.

Parsing a language input question s to an information descriptor z is equivalent to evaluating $P(z|s; D)$. We first process s by using Stanford CoreNLP Natural Language Parsing Toolkit [31] to extract part-of-speech (POS) tags, a context-free phrase structure tree, and results from applying co-reference resolution. We merge all outputs from the CoreNLP to a parse tree s' by copying the output parse tree and replacing its leaf nodes with the input words and the POS tag pairs, and then use the results from the co-reference resolution to replace the subtrees corresponding to the referring words with the subtree corresponding to the referred words. For example, given $s =$ "Is Mike Chung in his office?", the sentence extracted from s' is "Is Mike Chung in Mike Chung's office?".

Given s' we evaluate:

$$P((\tau, l, t)|s'; D) = \alpha \max_i \mathbf{1}(T_i^\tau(s')) \times \left\{ \max_j d(L(T_i^\tau(s')), A_j(l)) + \max_k d(G_i^\tau(s'), B_k(t)) \right\}. \quad (5)$$

where

- α is a normalization constant,
- $T_i^\tau(s')$ is an ith τ type relation template that can detect words describing a location and a target type in s'. Templates use relationships between tags (e.g. check if a node has children with tags PP and NP) and predefined keywords (e.g. check if a word paired with a IN POS tag equals to the locational preposition such as "in", "at", etc.) to detect the words. $\mathbf{1}(T_i^\tau(s'))$ returns a boolean variable indicating whether $T_i^\tau(s')$ fit on s' or not.
- $L(\cdot)$ and $G(\cdot)$ operators return detected words describing a location and a target type, respectively, from applying a template $T_i^\tau(s')$. Using the s' mentioned in the earlier example, $L(T_i^{\texttt{presence}}(s')) =$ "Mike Chung's office" and $G(T_i^{\texttt{presence}}(s')) =$ "Mike Chung" for some i.
- $A_i(l)$ returns ith words describing l and $B_j(t)$ returns jth words describing t by looking up the data stored in D. For example, $A_i(cse102) =$"Mike's Office" and $B_j(person) =$"Mike Chung" for some i, j.
- $d(\cdot, \cdot)$ function measures the similarity between two text inputs (e.g. Levenshtein distance).

If the input sentence is not a checking type question, then the distribution $P((\tau, l, t)|s'; D)$ will not be proper; no relation templates $T_i^\tau(s')$ can cover the input s'.

In Sec. IV-A, we approximate the summation in Eq. 2 with the max in Eq. 3. In other words, we are only considering the most likely information descriptor instead of all possible information descriptors.

VI. Viewpoint Estimation

As mentioned in Sec. IV-A, estimating the best viewpoint for answering the question asked by a user is equivalent to evaluating Eq. 4. In the following, we describe how we model the environment M and the terms involved in Eq. 4.

A. Environment Model

Our environment M is a tuple (M^{2D}, M^{3D}, M^T).

- M^{2D} is a 2D occupancy grid map with a resolution of 0.05m in which each grid cell is identified by its Cartesian coordinates in a global coordinate frame and described as either empty, occupied, or unknown. M^{2D} is acquired by mapping using a method developed by Grisetti et al. [32] and post-processed to only contain static information (e.g. walls and stationary furniture). M^{2D} is mainly used for navigation and for annotations in the database.
- M^{3D} is a 3D occupancy grid map similar to M^{2D} with an additional 3rd (height) dimension with a resolution of 0.05m. M^{3D} provides a richer representation of the environment; however, in dynamic environments it can quickly become outdated. Hence, we continuously update it with incoming depth data using Hornung et al.'s method [33]. M^{3D} is used for reasoning about visibility.
- M^T is a topological map in which each topological node is a candidate *place* for the viewpoints that the robot can gather information from. For each place, there is a discrete sets of candidate *orientations* that specify the *viewpoint*. When computing Eq. 1, we search for v^* in a M^T to make our problem tractable.

While we use existing mapping algorithms for acquiring M^{2D} and M^{3D} [32], [33], we use a custom algorithm for generating M^T as described in the following.

Topological Mapping. The topological map allows the system to constrain the problem of viewpoint estimation to a discrete subset of all possible viewpoints. This makes the problem tractable, but also results in a commitment that could harm performance. Therefore, it is important to select a discretization that properly supports the problem at hand.

We generate topological maps from a probability distribution $P(M^T|M^{2D})$ that models the relevance of locations to the task and distributes topological *places* accordingly. The Markov Random Field illustrating the distribution is shown in Fig. 2a and corresponds to:

$$p(N|M^{2D}) = \frac{1}{Z} \prod_i \phi_r(N_i)\phi_g(N_i), \quad (6)$$

where $N_i \in \{0, 1\}$ determines whether a place exists at location i and $N_i = \{N_j : j \in \text{neighborhood}(i)\}$ for a local spatial neighborhood of 1m radius.

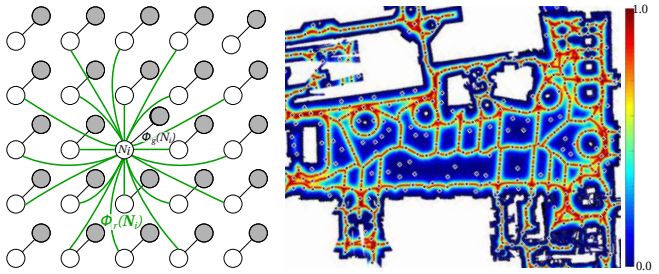

Fig. 2. Topological mapping. (a) Probabilistic graphical model illustrating the distribution from which topological maps are sampled. (b) A typical example of a set of topological nodes on top of the values of $\phi_g(N_i)$ for each pixel of an occupancy grid map.

The potential function $\phi_g(N_i)$ models the relevance of a location for the task and is defined in terms of three potentials calculated from the 2D metric map:

$$\phi_g(N_i) = \phi_o(N_i)\left(\phi_c(N_i) + \phi_v(N_i) - \phi_c(N_i)\phi_v(N_i)\right), \tag{7}$$

where:

- ϕ_o depends on the distance d_o to the nearest obstacle and is calculated similarly to the cost map used for the navigation algorithm [34]. ϕ_o equals 0 for distance smaller than the radius r of the robot base and $exp(-\alpha(d_o - r))$ otherwise.
- $\phi_v = exp(-\beta|d_o - d_v|)$ depends on the relation between the distance d_o and the fixed distance d_v that provides good visibility of obstacles in the map.
- $\phi_c = exp(-\gamma d_c)$ depends on the distance d_c to the nearest node of a Voronoi graph of the 2D map. This promotes centrally located places since central locations are often safe for navigation.

Overall, the definition of $\phi_g(N_i)$ ensures that candidate viewpoint locations are located only in areas that will not lead to a collision with obstacles and are either preferred due to their central location or visibility properties. The potential $\phi_r(N_i)$ ensures that places are distributed within certain distance to one another, by enforcing low probability for locations that are close to other existing places.

We employ Gibbs sampling to perform the maximum a posteriori inference and choose samples corresponding to maps with highest posterior probability. A typical example of a generated set of topological nodes for a single floor of a building is shown in Fig. 2b. For each place, we assume a discrete set of *orientations* evenly spread across the full circle. The orientation and the metric position of a place fully specify a *viewpoint*. The resulting map M^T is expressed as a set of viewpoints M_i^T, with each view anchored in a metric map M^{2D}.

B. Information Presence Term

The first term in Eq. 4, $P(\mathcal{I} = 1|y, z; M, D)$, models the presence of the information specified by z at location y (a cell in M^{2D}). It is computed based on annotations provided a priori and stored in the database D. Annotations are associated with polygon regions (e.g. the room annotation in Fig. 3a) or a set of discrete points (e.g. annotation of *person* in Fig. 3b) on M^{2D}. They are divided into two groups; one corresponding to a location name (l in z) and another corresponding to the presence of the target type in the specified region (t in z). The location name annotations are associated with polygon regions and have the same value $P(\mathcal{I} = 1|y, z; M^{2D}) = 1$ at all cells y within the specified region. On the other hand, the target type annotations are associated with a discrete set of points with non-zero probability where they are likely to be present (e.g., an object near table, a person near desks). Assuming independence between two groups, we have $P(\mathcal{I} = 1|y, z; M^{2D}) = P(\mathcal{I} = 1|y, l; M^{2D})P(\mathcal{I} = 1|y, t; M^{2D})$.

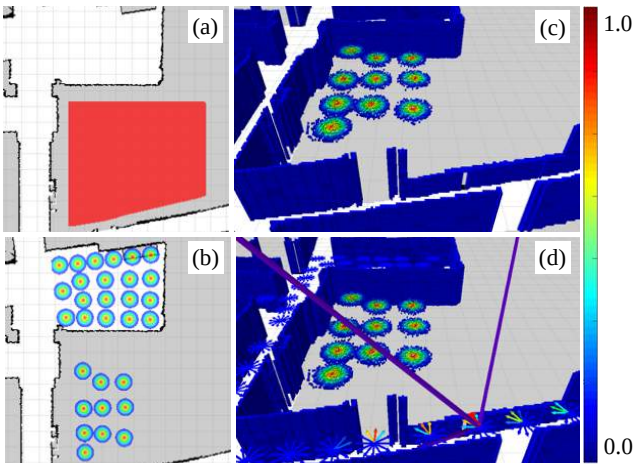

Fig. 3. Annotations, information presence term, and viewpoints. (a) and (b) display respectively a room and person annotations overlaid on M^{2D}. (c) shows the information presence term computed using (a) and (b). (d) shows the evaluated viewpoints $P(v, \mathcal{I} = 1|z; M, D)$ as colored arrows and the camera field of view of the optimal viewpoint drawn with the purple lines.

We transform $P(\mathcal{I} = 1|y, z; M^{2D})$ to $P(\mathcal{I} = 1|x, z; M)$ using a $2D$ to $3D$ coordinate conversion function $f|M^{3D} : Y \rightarrow X$. The function $f|M^{3D}$ maps an input 2D coordinate to a 3D coordinate by extending the input 2D coordinate with a height value. Assuming the objects of interest are usually located at a certain distance above the ground (e.g. people), we sample the height value from a Gaussian distribution with the mean μ and the standard deviation σ.

C. Visibility Term

The second term in Eq. 4, $P(x|v; M^{3D})$, models the visibility of cells x in M^{3D} from viewpoint v. We compute it using raytracing in M^{3D}. We set $P(x|v; M) = 0$ for (i) x that are not visible and (ii) x that are located farther than θm from the camera origin. We set $P(x|v; M) = 1.0$ for all x that are within the camera's cone of visibility. We experimented with other $P(x|v; M)$ such as $P(x|v; M)$ dependent linearly on the Euclidean distance between x and the camera origin; however, the performance differences were negligible in our experimental setting. Finally, while the robot is executing the task, it might discover certain v are not reachable. For those v, we set $P(x|v; M) = 0$ all x.

D. Iterative Refinement

Once v^* is computed based on Eq. 1, the robot starts navigating to v^*. Since the robot is operating in a dynamic

environment, M in Eq. 1 might be outdated. As a result v^* may not provide the best viewpoint with respect to the new M. Assuming the availability of the component that can track the changes in M with incoming data, we address this problem by letting the system continuously re-evaluate $P(\mathcal{I} = 1|v, z; M, D)$ until there is no change in v^*. Note that we do not re-compute $P(z|s; D)$ in Eq. 2 since it is not dependent on M. Once the robot reaches the final v^*, it saves an image from the on-board camera (Fig. 1 (c)) and returns this image as its response (Fig. 1d).

VII. Experiments and Results

A. Natural Language Parsing

We evaluated our input question parsing component on the real user questions collected during the deployment experiment described in Sec. III. The labels for the questions were acquired by a coding process performed by two of the authors. Labeling involved writing an information descriptor z^* for each question sentence s as $z^* = \text{argmax}_z P(z|s; D)$. For the locations l in information descriptors $z = (\tau, l, t)$ we used the 295 unique locations extracted from the building database.

In order to test our system's ability to correctly parse questions that involve information checking, we first ran a checking vs. non-checking classification experiment. Our system classifies a question as "checking" if the output distribution from the parser is proper (i.e. $\sum P(z|s, D) = 1$), and as "non-checking" otherwise. We attained an accuracy of 74%, a precision of 94% and a recall of 71% (# of true positives: 48, true negatives: 17, false positive: 3, false negatives: 20). A high precision rate is desirable from the robot's perspective as false positives will result in executing the wrong task. An example of a false positive is "What does Mike Chung look like when he's not at his desk?" (negation "not"). Examples of false negatives include "How many LEDs are on the wall in the Atrium?" (two locational PP phases), "Is the service elevator in the CSE building operational?" (implicit/unknown target location), and "What color is Hank wearing today?" (implicit/unknown target location).

For the 65 questions that were correctly identified as checking questions, we evaluated our system's ability to extract the corresponding information descriptor. Our parser achieved an accuracy of 95% in classifying the question subtype τ, 89% in classifying the location l, 82% in classifying the target type t, and 72% in correctly classifying the full information descriptor $z = (\tau, l, t)$.

B. Viewpoint Estimation

We evaluated our viewpoint estimation component with two experiments involving the real robot in the computer science department building.

Experimental Setup. We used the custom-built mobile robot based on the MetraLabs Scitos G5 mobile base expanded with a structure providing support for sensors and user interfaces (Fig. 1b and Fig. 1c). A high-resolution Allied Vision Manta G609 camera with 97° horizontal and 79° vertical view angle, which is used for providing images to the users, is attached to the robot at 1.31m above the ground. An Asus Xtion Pro depth camera is placed at 1.25m above the ground to collect depth images for the purpose of building 3D maps. Another backward facing Xtion depth camera and a Hokuyo UTM-30LX laser range finder are also placed on-board for navigation purposes.

The 2D occupancy maps M^{2D} used in our experiments were collected prior to running experiments. The initial 3D occupancy maps M^{3D} were constructed from the corresponding 2D occupancy maps by extending occupied cells to the default wall height (2m) or default window height (0.85m) depending on their location. For topological map M^T generation, we used the following parameter values: $\alpha = 5$, $\beta = 8$, $\gamma = 10$. The discrete orientations of the viewpoints were distributed in the topological map every 30°. For all maps, we only represented the open spaces such as corridors and breakout areas to avoid going into building occupants' offices during working hours.

The 295 location annotations were imported from the building database and the person target type annotations were acquired by a manual annotation process. These annotations were transformed to the information presence term using the height-based conversion function with the height mean $\mu = 1.65$ and standard deviation $\sigma = 0.05$. For the visibility term, the horizontal and vertical field of view of the camera were discretized into a 100×100 grid for raytracing, and used $\theta = 15$.

1) Experiment I: We evaluated our viewpoint estimation algorithm's ability to deliver images that can be used for answering checking questions on questions frequently asked during the initial deployment (Sec. III):

- Q1. *Is {person} in his/her office?*
- Q2. *Is there anyone in the mobile robotics lab?*
- Q3. *Is the breakout area occupied?*
- Q4. *Is the conference room occupied?*
- Q5. *Is there a stapler in the printer room?*[1]

Note that all questions were yes/no questions. The corresponding best fit information descriptors for the questions are shown as the column headers in Fig. 4. We ran the viewpoint estimation with the iterative refinement four times throughout a day for each checking question. We choose the timing of the run so that the ground truth answer for two runs were "yes" and the other two runs were "no". However, we did not control the visibility and reachability conditions of the target locations to capture natural variations in the building environment. Fig. 4 shows the returned images from each run and Fig. 6 shows the details of the viewpoint estimation algorithm for Q2 and Q3 runs.

Viewpoint quality. To understand potential users' ability to extract answers from images chosen by our viewpoint estimation methods, we conducted a user study with 10 building occupants. For each image returned from the runs

[1]Equivalent to "Is there free food in the kitchen?"

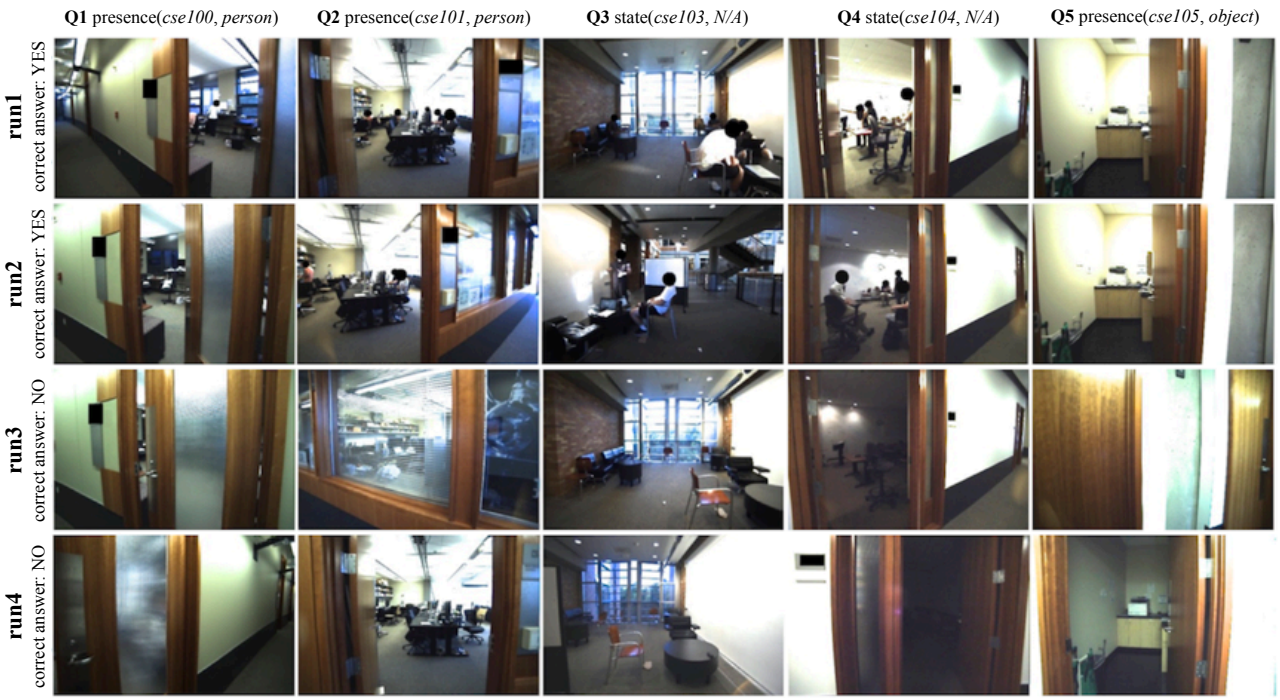

Fig. 4. Returned images from the experiment I runs. The first two row show images from the runs with the ground truth answer "yes" and the next two rows show images from the runs with the ground truth answer "no" for the corresponding checking questions (columns). The column header displays the best fit information descriptors for the corresponding checking questions (Q1–Q5).

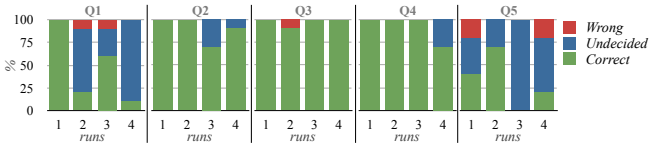

Fig. 5. Distribution of answers generated by user study participants from images captured by the robot.

described above, we asked participants to respond to the corresponding checking question (Q1-Q5) based on the image. Response options were "definitely yes", "probably yes", "I don't know", "probably no", and "definitely no". Fig. 5 shows the results from the user study. We consider a response to be correct if a user responds with "definitely yes" or "probably yes" when the ground truth answer is "yes" or if they say "definitely no" or "probably no" when the ground truth answer is "no". We consider a response wrong if the user's answer contradicts the ground truth and undecided if the user responds with "I don't know."

Overall participants achieved a high classification accuracy, particularly for questions Q2, Q3, and Q4. It can be observed from Fig. 4 that high "I don't know" rates and non-zero wrong response rates are due to the limitations of the sensors or the encountered situation (e.g. target locations blocked by closed doors or bad lightening conditions) rather than a limitation of the algorithm. For example, in the 4th run for Q1, 90% of participants said they were undecided if there is a person inside the office because they observe the door being closed; not because the robot did not provide sufficient information. In other words, if the users were to try and answer Q1 in this situation by going to the target location themselves, they would reach the same answer through passive observation. Similarly in the third run of Q5, all of the participants indicated that they did not know if there was a stapler in the printer room because the door to the room was closed.

In the other runs of Q5, the wrong and undecided answers are due to the difficulty of seeing the stapler in the small and reduced-quality image (due to lighting). This problem could be mitigated by allowing the robot to navigate into the room to obtain a better viewpoint, post-processing images to enhance color contrast, or allowing participants to zoom in on parts of the image to obtain the answer.

Handling dynamic changes. Fig. 6 illustrates how the viewpoint estimation algorithm adapts to dynamic changes in the environment by selecting alternative viewpoints with similar information content. For Q2, the robot was able to capture the view of the lab's inside through its door when it was open, but also through its window when the door was closed and the blinds on the window were open (run 3). Similarly for Q3, the robot navigated to the other end of the breakout area and turned around to capture the view of the area, when it encountered a whiteboard blocking the view from its initial viewpoint.

2) Experiment II: Next, we considered the retrospective querying scenario in which the viewpoint estimation is performed on previously collected data. This captures scenarios that involve questions concerning the past ("Was Mike Chung in the robotics lab?") where the system attempts to provide a response based on incidental visits to a place while performing other tasks involving navigation (e.g. patrolling or delivery). We considered four such cases. In the first two, the robot was navigating near the breakout area as shown in

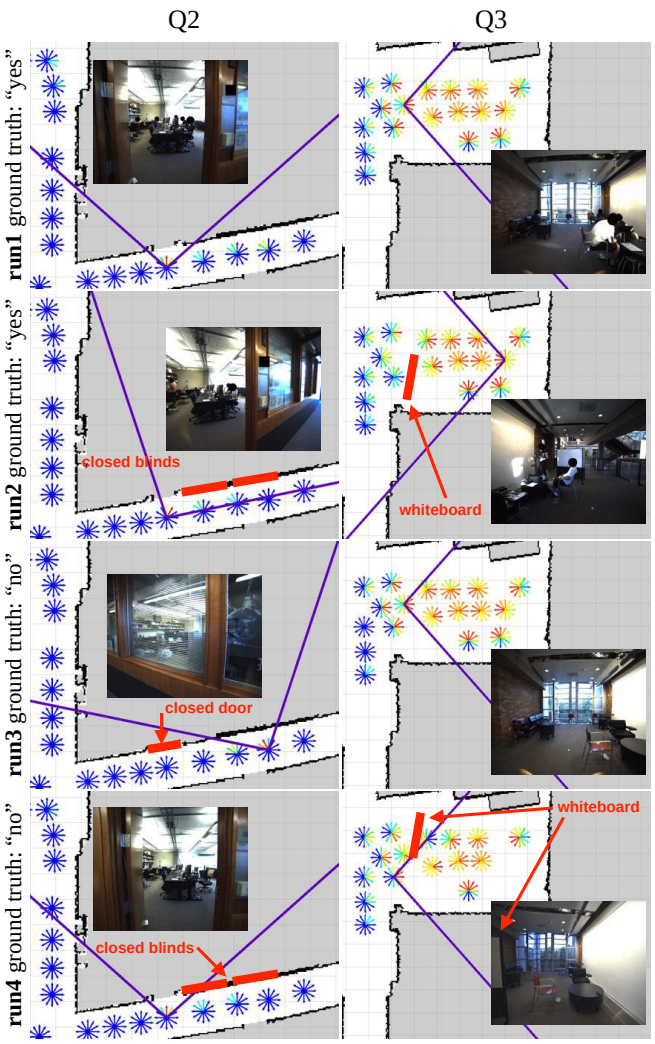

Fig. 6. Viewpoint estimation details for Q2 and Q4 in Experiment I. The evaluated quality of each viewpoint, based on $P(\mathcal{I} = 1|v, s; M, D)$, are displayed as colored arrows over the 2D map. Warm colors (red) indicate greater quality. The camera field of views of the selected optimal viewpoint are drawn with the purple lines and the image captured from this viewpoint is shown. The dynamic changes to the environment that influenced the viewpoint estimation algorithm are annotated in red.

the left column of Fig. 7 and the viewpoint estimation was later used to answer the question Q6: "Was the breakout area occupied?". In the latter two cases, the robot was navigating near the conference room as shown in the right column of Fig. 7 and the viewpoint estimation was later used to answer the question Q7: "Was the conference room occupied?". In all cases, the viewpoint estimation algorithm was used with the latest 3D occupancy map available from the collected data, within the constrained search space of *visited* viewpoints.

The retrieved images and details of the runs for this experiment are shown in Fig. 7. We observe that the viewpoint estimation algorithm produces appropriate responses in the retrospective question answering setting. In response to Q6, the robot needs to capture the breakout area from a set of candidate viewpoints that are tangential to the area (i.e. the robot went by the breakout area without looking towards it). We see that the algorithm selects viewpoints that are

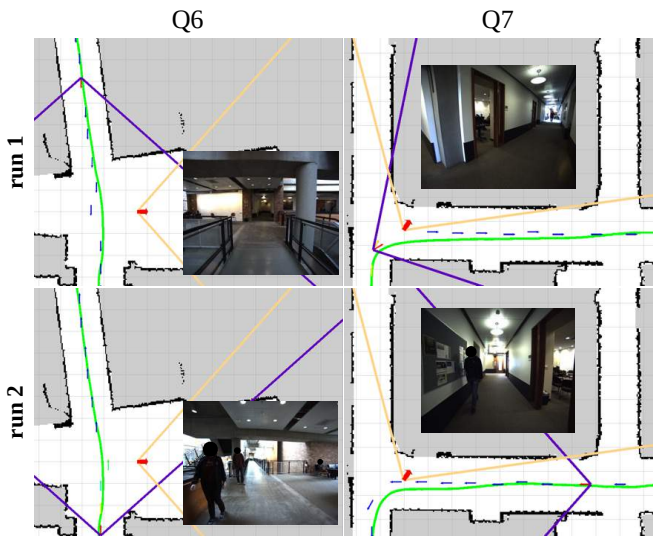

Fig. 7. Experiment II results. The path taken by the robot is shown with the green line and the evaluated quality of viewpoints along this path are displayed as colored arrows. Warm colors (red) indicate greater quality. The camera field of view of the selected viewpoint on the path is shown with the purple lines and the image captured from this viewpoint is provided. For reference, the optimal viewpoint that would have been selected if the robot were to navigate back to the scene to capture the requested information is shown with the orange lines.

further away in the path such that the target area can be captured on one side of the robot's field of view. In run 1, the robot is able to choose viewpoints that are further from the target area than in run2, and hence captures more of the area by exploiting the fact that the bridge-like corridor does not have walls obscuring the robot's view of the target area. In response to Q7, the robot is able to capture a larger part of the target conference room by choosing viewpoints near two different doors to the room in the two different runs where the robot was navigating in opposite directions.

VIII. Discussion

We tackled the problem of using mobile robots to check information about the environment in order to answer a question asked by a user. Overall, our findings indicate that the proposed framework and the implementation presented in this paper address the problem reasonably well. Our previous work motivated the usefulness of this capability from an end-user perspective [1], while this paper demonstrates the feasibility of autonomously providing this capability.

There are nonetheless several assumptions made in scoping our problem and some limitations to the proposed approach. First, we assume that the user's question mentions a single target location that can be feasibly captured from a single viewpoint. We consider a question such as "Is Mike Chung in this building?" as a *search* type question, and therefore out of scope for our framework. However, one can imagine a search method that embeds our approach for checking information at multiple target locations, within a larger planning framework. Similarly questions that mention multiple target locations, such as "Is Mike Chung in the robotics lab or his office?" are not handled by our natural language component; however, this task could simply be

considered as two separate information checking requests.

Another limitation is that the human users need to extract the answer to their own question from the provided image, rather than receiving a definite answer. Although this part of the task could also be automated with recent image understanding methods, we chose to leave it to the users since they can perform image understanding tasks robustly and efficiently [35]. Finally our work focused on capturing information from a single image while images from multiple viewpoints or multiple images from the same viewpoint (to capture dynamic events) could potentially provide answers to a richer set of questions.

IX. Conclusion

We present a framework and an end-to-end system for answering natural language questions from users about the robot's environment. The robot parses the question into an information request and computes the best viewpoint for capturing the information, based on a semantically annotated map of the environment. It then navigates to this viewpoint, while re-computing the optimal viewpoint based on the most up-to-date state of the environment, and returns an image that contains the answer to the user's question. We evaluate our system with questions collected from diverse inhabitants of the building in which the robot is deployed. We demonstrate our system's ability to parse these questions and choose appropriate viewpoints to answer them in different states of the environment. We also demonstrate that our method can be used for answering question retrospectively, by selecting a previously recorded image to answer a question.

References

[1] M. Chung, A. Pronobis, M. Cakmak, D. Fox, and R. P. Rao, "Designing information gathering robots for human-populated environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[2] M. Veloso, J. Biswas, B. Coltin, S. Rosenthal, T. Kollar, C. Mericli, M. Samadi, S. Brandao, and R. Ventura, "Cobots: Collaborative robots servicing multi-floor buildings," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[3] "Aethon TUG," http://www.aethon.com/tug/, [Online; accessed 07-25-2015].

[4] "Savioke SaviOne," http://www.savioke.com, [Online; accessed 07-25-2015].

[5] "Vecna QC Bot," http://www.vecna.com/product/qc-bot-base-model/, [Online; accessed 07-25-2015].

[6] G. Kim, W. Chung, K.-R. Kim, M. Kim, S. Han, and R. H. Shinn, "The autonomous tour-guide robot jinny," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[7] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot," in *IEEE International Conference on Robotics and Automation*, 2003.

[8] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, *et al.*, "MINERVA: A second-generation museum tour-guide robot," in *IEEE International Conference on Robotics and Automation*, 1999.

[9] A. Ohya, Y. Nagumo, and Y. Gibo, "Intelligent escort robot moving together with human-methods for human position recognition," in *International Conference on Soft Computing and Intelligent Systems*, 2002.

[10] M. K. Lee, S. Kiesler, and J. Forlizzi, "Receptionist or information kiosk: how do people talk with a robot?" in *ACM Conference on Computer Supported Cooperative Work*, 2010.

[11] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "An affective guide robot in a shopping mall," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2009.

[12] J. Park and G. J. Kim, "Robots with projectors: an alternative to anthropomorphic hri," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2009.

[13] M. Lorbach, S. Hofer, and O. Brock, "Prior-assisted propagation of spatial information for object search," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[14] A. Aydemir, A. Pronobis, M. Gobelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *IEEE Transactions on Robotics*, 2013.

[15] M. Samadi, T. Kollar, and M. M. Veloso, "Using the web to interactively learn to find objects." in *AAAI Conference on Artificial Intelligence*, 2012.

[16] L. Kunze, M. Beetz, M. Saito, H. Azuma, K. Okada, and M. Inaba, "Searching objects in large-scale indoor environments: A decision-theoretic approach," in *IEEE International Conference on Robotics and Automation*, 2012.

[17] D. Joho and W. Burgard, "Searching for objects: Combining multiple cues to object locations using a maximum entropy model," in *IEEE International Conference on Robotics and Automation*, 2010.

[18] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *IEEE International Conference on Robotics and Automation*, 2009.

[19] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation." in *AAAI Conference on Artificial Intelligence*, 2011.

[20] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2010.

[21] J. Fasola and M. J. Mataric, "Interpreting instruction sequences in spatial language discourse with pragmatics towards natural human-robot interaction," in *IEEE International Conference on Robotics and Automation*, 2014.

[22] V. Perera and M. Veloso, "Handling complex commands as service robot task requests," 2015.

[23] A. Jacoff, "Search and rescue robotics," in *Springer Handbook of Robotics*, 2008.

[24] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff, "Autonomous and autonomic systems: A paradigm for future space exploration missions," *Transactions on Systems, Man, and Cybernetics*, 2006.

[25] J. Casper and R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics*, 2003.

[26] R. Bachmayer, S. Humphris, D. Fornari, C. Van Dover, J. Howland, A. Bowen, R. Elder, T. Crook, D. Gleason, W. Sellers, *et al.*, "Oceanographic research using remotely operated underwater robotic vehicles: Exploration of hydrothermal vent sites on the mid-atlantic ridge at 37 north 32 west," *Marine Technology Society Journal*, 1998.

[27] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *International Journal of Robotics Research*, 2014.

[28] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *International Journal of Robotics Research*, 2012.

[29] J. Velez, G. Hemann, A. S. Huang, I. Posner, and N. Roy, "Planning to perceive: Exploiting mobility for robust object detection." in *International Conference on Automated Planning and Scheduling*, 2011.

[30] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, 2011.

[31] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014.

[32] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, 2007.

[33] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[34] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon," 2010.

[35] M. C. Potter, B. Wyble, C. E. Hagmann, and E. S. McCourt, "Detecting meaning in rsvp at 13 ms per picture," *Attention, Perception, & Psychophysics*, 2014.