2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Robot Object Referencing through Legible Situated Projections

Thomas Weng[1], Leah Perlmutter[2], Stefanos Nikolaidis[3], Siddhartha Srinivasa[2], and Maya Cakmak[2]

***Abstract*— The ability to reference objects in the environment is a key communication skill that robots need for complex, task-oriented human-robot collaborations. In this paper we explore the use of *projections*, which are a powerful communication channel for robot-to-human information transfer as they allow for situated, instantaneous, and parallelized visual referencing. We focus on the question of what makes a *good* projection for referencing a target object. To that end, we mathematically formulate *legibility* of projections intended to reference an object, and propose alternative arrow-object match functions for optimally computing the placement of an arrow to indicate a target object in a cluttered scene. We implement our approach on a PR2 robot with a head-mounted projector. Through an online (48 participants) and an in-person (12 participants) user study we validate the effectiveness of our approach, identify the types of scenes where projections may fail, and characterize the differences between alternative match functions.**

I. Introduction

Robots are entering new environments that require constant communication with human collaborators about task-relevant information. In particular, many joint human-robot tasks require the robot to reference an object in the environment to provide information about it or instruct the human to perform an action with it. Much previous work on robotic spatial referencing focuses on speech, gaze, gestures, and secondary displays. However, these methods have limitations with precision and speed. Verbal descriptions take time to utter, can have ambiguity inherent to natural language, and might sound unnatural. Gesturing and gazing also take time and can have high ambiguity depending on the robot's embodiment. Visualizations on a screen can accurately indicate an object but are not situated in the task context and require a mental mapping.

Projections resolve most of these limitations. Compared to other communication channels, projections can make faster, parallelized, more precise, and more intuitive references. Projectors are small and power-efficient enough to be mounted on robots as portable displays.

While previous work demonstrates the potential benefits of projections (Sec. II), the question of what constitutes a *good* projection for human-robot communication remains open. Our work aims to tackle this question, starting with

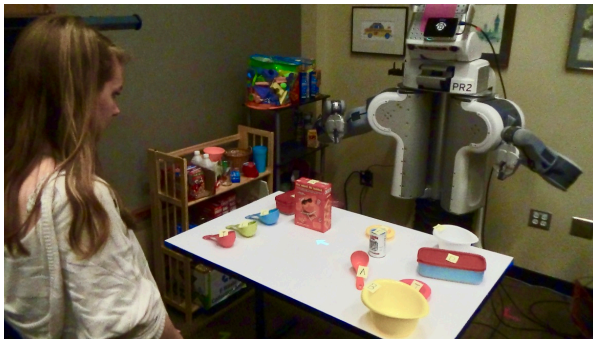

Fig. 1: We propose a mathematical framework to select where a robot should project an arrow on a cluttered tabletop to reference a target object during human-robot collaboration.

communication of object references. In this paper we explore how a robot should project an indicator to single out a target object on a cluttered tabletop environment. We introduce a mathematical model for legible placement of projected object references for tabletop objects. We focus on the use of arrows to reference objects and propose an arrow model that estimates the probability that a projected reference indicates an object given its configuration relative to the object. We use this framework to optimally select indicators for a target object in synthetic and real world scenes. Our model accounts for occlusions to produce a visible placement of a projection that clearly references an object. We evaluate our model's performance in two user studies: an online study with 2D synthetic scenes and an in-person study with the PR2 robot referencing objects on a table top. Our studies demonstrate that our method can select effective arrows in most scenarios. We also characterize the differences between alternative arrow-object match functions used in our optimization.

II. Related Work

Projections in robotics. Projections have been used previously for human-robot communication. Andersen *et al.* present a system for tracking objects, such as a car door being assembled or a box on a table, in human-robot collaboration tasks and projecting task-related information directly onto the object and on the workspace [1]. In their user study the robot projects a destination area for where an object needs to be moved by the person, arrows that indicate how an object should be rotated, and iconic symbols such as a warning sign or a checkmark that indicate task status. In Chadalavada *et al.* a mobile robot projects its planned path onto the floor [2]. In Nguyen *et al.* the human partner uses a laser pointer to project a dot of light on an object for the robot to retrieve

[1]Thomas Weng is with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA tweng@andrew.cmu.edu

[2]Leah Perlmutter, Siddhartha Srinivasa, and Maya Cakmak are with the Paul G. Allen School of Computer Science and Engineering, University of Washington, 185 E Stevens Way NE, Seattle, WA 98195, USA {lperlmu,siddh,mcakmak}@cs.washington.edu

[3]Stefanos Nikolaidis is with the Department of Computer Science, University of Southern California, 941 Bloom Walk, Los Angeles, CA 90089, USA nikolaid@usc.edu

[3]. In Lazewatsky *et al.*, a robot projects a cursor where the robot thinks the human is facing [4]. While these works explore different ways of using projections, we focus on object referencing and ways to optimize references.

Spatial referencing. Beyond projections, many researchers in the HRI community have studied spatial referencing of objects or people by the robot using speech [5], [6], pointing [7], [8], [9], and gaze [10]. Roy *et al.* developed an algorithm to detect and resolve spatial ambiguities with speech [11]. Admoni *et al.* and Stiefelhagen *et al.* combine speech, gaze, and pointing to disambiguate object references [12], [13].

Legibility. Dragan *et al.* characterize predictability and legibility in the context of a robot reaching for an object. A predictable motion is the lowest cost motion to reach the goal, but may be ambiguous to an observer. A legible motion reaches the goal less efficiently while using more energy to broadcast the reacher's intent to an observer [14]. Holladay *et al.* study the tradeoff between clarity and efficiency when a robot makes pointing gestures [7]. Our work applies this notion of legibility to projection-based communication.

III. Referencing Objects with Projections

Human-robot collaboration on joint tasks can involve communication of many types of information in both directions. Our work focuses on the communication of a single target object by the robot to the human. For example, consider a social robot on a kitchen counter that guides its user through a recipe. At every step of the recipe, the robot needs to instruct the user to add a particular ingredient or use a particular tool that might be on the user's workspace. The robot could use projections to unambiguously indicate these items to the user. As another example, consider a robot co-worker in a factory setting collaborating with a human to transfer objects from the workspace to a package. The robot could indicate which object it is going to pick up next as a way to increase the human's awareness of what the robot is doing and improve the fluency of the collaboration.

There are several ways to use projections for indicating objects, such as directly projecting onto the object, encircling the object with the projection, or placing an indicator near the object. We focus on the latter option due to its greater generalizability to objects of various shapes, sizes, and colors. We also choose *arrows* specifically to take advantage of their directionality. The key question we tackle in this paper is how to chose an arrow to indicate a particular object in a given scene.

A. Choosing Legible Arrows

To formalize the object referencing task, let $\mathcal{O}$ be the set of all candidate objects that reside on surface R, with $\|\mathcal{O}\| = n$. Let $o^* \in \mathcal{O}$ be the *target object* and $\mathcal{O}^D = \mathcal{O} - o^*$ be the set of *distractor objects*. We define an arrow α as a tuple (x, y, θ), with $(x, y) \in \mathbb{R}^2$ and $\theta \in [0, 2\pi]$.

In referencing a target object with a projected arrow, the robot's goal is to choose values of α that maximize the probability that the target object is being referenced. We write this optimization problem as choosing an arrow that maximizes the probability that the target object is being indicated given the arrow.

$$\alpha^* = \arg\max_{\alpha} P(o^*|\alpha) \tag{1}$$

According to the Bayes rule we can rewrite the term being maximized as

$$\frac{P(\alpha|o^*)P(o^*)}{P(\alpha)} = \frac{P(\alpha|o^*)P(o^*)}{\sum_{o_i \in \mathcal{O}} P(\alpha|o_i)P(o_i)}$$

We assume that all objects have equal priors, $P(o^*) = P(o_i) = \frac{1}{n}$. Hence, our ability to compute Eqn. 1 depends on estimating the probability of an arrow given a target object. To that end, we define a probability distribution over arrows as $P(\alpha|o) \propto e^{-d(\alpha,o)}$ where the function d is a distance function defined between an arrow (x, y, θ) and an object. As a result we can rewrite Eqn. 1 as

$$\alpha^* = \arg\max_{\alpha} \frac{e^{-d(\alpha,o^*)}}{\sum_{o_i \in \mathcal{O}} e^{-d(\alpha,o_i)}} \tag{2}$$

Matching our intuition of what a legible arrow should do, this computation will maximize the match between the arrow and the target object, while minimizing the match to the distractors.

B. Modeling the Background

Assume there is only one object in the scene. The term being maximized in Eqn. 2 then becomes a constant (*i.e.,* 1), independent of the arrow. In other words, if there is only one object, *any* arrow is considered to be pointing at it. In reality, it is counterintuitive for an arrow pointing in the opposite direction of an object to be considered as pointing at it. To capture this intuition in our model, we assume a *background* object $o_0 \in \mathcal{O}$ that is always present. In the simplest case we can assume a background which has a constant probability of being pointed at independent of the arrow configuration, *i.e.,* $P(\alpha|o_0) = p_0$. As a result, the denominator of Eqn. 2 will always have a constant term and the maximization will force the arrow to point towards the object, even when there is only one object.

C. Modeling Arrow-Object Distance

Next we need to define $d(\alpha, o_i)$ in Eqn. 2 to capture the space of arrow-object relationships, *i.e.,* how well the arrow indicates the object. This distance should be low if the arrow is highly indicative of the object and it should be high otherwise. In the following we identify four intuitive arrow-object distance functions, based on different object representations. The first function is focused on the relative *direction* of the arrow while the other three are focused on the relative *location* of the arrow. Ultimately a combination of these functions will enable fully specifying an optimal arrow placement (Sec. III-C.5).

1) Relative angle: Intuitively an arrow is likely to be perceived as pointing at an object if its direction intersects with the object. The match between an arrow and an object should be higher if the arrow is pointing towards the center of the object (as compared to its edges). To formulate a simple arrow-object distance function that captures this intuition, assume that the object o_i is represented as a point (x_i, y_i) on surface R. Then, let our first arrow-object distance function be defined as:

$$d_1(\alpha, o_i) = \left\| \arctan(\frac{x - x_i}{y - y_i}) - \theta \right\|$$

This function measures the angle between the arrow direction and the direction of a vector that connects the tip of the arrow to the location of the object.

2) Proximity: We also expect the match between an arrow and an object to be higher when the two are close to one another. To capture this intuition we consider the following distance function that measures the Euclidean distance between the arrow and the object:

$$d_2(\alpha, o_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

3) Edge proximity: The two distance functions above do not depend on the shape or size of the object, whereas in practice those might impact the quality of match between an arrow and object. For example, consider a long and slim rectangle—an arrow that is pointing to it near its short edge might not be considered much worse than one that is pointing at its long edge, since the arrow is close to the edge of object, even though it might be far from its center hence worse according to d_2. To capture this difference we define an alternative distance function which is the shortest distance from the arrow tip to the edge of an object. Although this distance can be computed analytically if the object geometry is known, in the general case we assume the object o_i contains a set of points O_i on the projectable region R and define the function as follows:

$$d_3(\alpha, o_i) = \min_{x_{ij}, y_{ij} \in O_i} \sqrt{(x - x_{ij})^2 + (y - y_{ij})^2}$$

4) Object span: An arrow that is at a particular proximity to the center or closest edge of an object might still differ in how well it indicates the object depending on the size of the object. The larger the object, the more "buffer" it will have around the pointed direction to prevent misinterpretation of the arrow. We capture this insight with the following distance function that measures the *span* of the object from the perspective of the arrow:

$$d_4(\alpha, o_i) = \left\| \max_{x_{ij}, y_{ij} \in O_i} \arctan(\frac{x - x_{ij}}{y - y_{ij}}) - \min_{x_{ij}, y_{ij} \in O_i} \arctan(\frac{x - x_{ij}}{y - y_{ij}}) \right\|^{-1}$$

This function jointly captures the proximity of the arrow to the object and the size of the object. Given a fixed arrow, the object-arrow match increases with closer and larger objects.

5) Combining distance functions: Although some other distance functions are possible, the set above captures several key intuitions about measuring the match between an arrow and an object. Relative angle (d_1) will find an arrow that points as much as possible towards the target object and away from the distractors. In the absence of distractors, this function will only influence the orientation (θ) of the arrow; however, when distractors are present, different positions (x, y) of the arrow will result in different relative angles for each object. Hence, this function alone can be used for fully specifying an arrow (x, y, θ). In contrast, d_2, d_3, and d_4 will only influence the position of the arrow; hence, they need to be combined with d_1 to also specify orientation. Therefore we define the following functions: $d_A = d_1$; $d_B = d_1 + d_2$; $d_C = d_1 + d_3$; and $d_D = d_1 + d_4$.

6) Ray-based distance approximation: We propose an additional way of jointly optimizing the orientation and position of the arrow based on the legible pointing method proposed by Holladay *et al.* [7]. This involves integrating the *deviation* from the arrow orientation *over the span of the object*. Like Holladay *et al.* we use a numerical approximation of this integration. We consider a finite set of rays from -90^o to 90^o around the arrow, where 0^o corresponds to the direction of the arrow. Each ray has a weight w_β that reflects how far the ray deviates from the arrow. We use $w_\beta(\beta_i, \alpha) = e^{-(\theta - \beta_i)^2}$ where β_i is in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Then we define the distance function d_E as follows:

$$d_E(\alpha, o_i) = \sum_{\beta_j \in [-\frac{\pi}{2}, \frac{\pi}{2}]} w_\beta(\beta_i, \alpha) I(\beta_i, \alpha, o_i)$$

The indicator function $I(\beta_i, \alpha, o_i)$ is 1 if the ray in the direction $\theta - \beta_j$ intersects with the object o_i and 0 otherwise. Just like the object span distance function (d_4) proposed earlier this will favor arrow positions closer to the target to have more rays intersect with the object; however, it will also push the arrow to point towards the target object to maximize the summed weight.

D. Computing Optimally Legible Arrows

The optimization problem at hand is to search for arrow parameters α maximizing the conditional probability stated in Eqn. 2 with different combinations of distance functions. Given the complexity of some of our cost functions, computing an analytical gradient is not an option. Further, in real world projections, we need to take into account certain constraints related to projectability and visibility. To address these challenges we use the *constrained optimization by linear approximation* (COBYLA) algorithm [15] which is a gradient-free optimization method. In this paper we used the COBYLA implementation available in the NLopt package [16]. Because this approach is susceptible to starting configurations, we randomly sample multiple starting configurations to get a robust result.

1) Optimization Constraints: The optimization is constrained to place arrows within the projectable and visible space on a projection plane. The space of possible projections is bounded by the range of the projector and its pose relative

to the projection plane. It is further narrowed with the space covered by objects, fall under the object's shadow from the perspective of the projector, or behind the object therefore not visible to the human (Fig. 1). To avoid cutting off parts of the arrow due to sensor noise, a certain buffer around these regions should also not be used for projections. In the optimization process, we avoid placing arrows in the these regions by associating a high cost with them.

In this work, we assume that the human is located on the opposite side of the projection plane from the robot and compute the constraints based on a fixed human viewpoint. While visibility of the arrow from the user's perspective is a hard constraint that needs to be satisfied, we think that arrows closer to the user might be preferable over further ones for providing higher visibility, since the size of the arrow is kept constant in this work. Hence we also consider adding the following term to the optimization function, where (x_h, y_h) is the position of the human in the projection plane and w_h is a constant for balancing the effect of this additional constraint.

$$w_h e^{-\sqrt{(x-x_h)^2+(y-y_h)^2}} \tag{3}$$

The impact of this term was evaluated in lab studies with users across the table from the robot.

E. Robot System Implementation

We implemented our system on a PR2 robot.

1) Perception: We used the PR2's head-mounted Kinect device for scene perception. PCL and OpenCV were used for table-top segmentation [17], [18], which provided surface coordinates while segmenting objects on the surface. Based on the known coordinates and parameters of the projector as well as the assumed pose and visual field of the user, this map was pruned to only include projectable space that was un-occluded from the user's point of view.

2) Projection: We mounted a 500-lumen portable projector on the robot's pan-tilt head. The projector produces no light for pure black image pixels, enabling us to project selectively. The projector was calibrated to project onto the table-top of the in-person evaluation discussed in Sec. IV-C. Our projection system receives the positions of detected scene objects, the map of projectable, un-occluded regions, and a target object. The optimization described in Sec. III-C fits an image of an arrow's position and orientation, relative to the target object given the constraints of the map. The arrow was rendered by superimposing a small image of a standard arrow at the correct pose on a black image on the surface plane. Finally, we homographically transform the arrow image from the table plane to the projector plane to obtain the image that the projector should display so that the arrow appears flat on the table.

IV. Evaluation of Legible Projections

The legibility of arrows produced with our approach needs to be evaluated empirically from the users' perspective, since there is no ground truth arrow placement. In the following we first present outcomes of our approach in example scenes

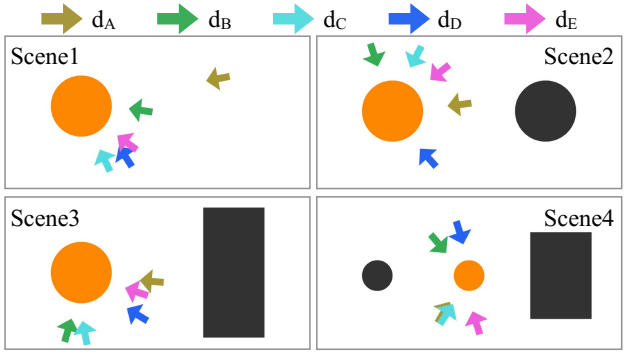

Fig. 2: Arrows chosen with the five different object-arrow distance functions (d_A: angle only, d_B: angle+proximity to object center, d_C: angle+edge proximity, d_D: angle+object span, d_E: weighted ray overlap) in four example scenes where the orange object is the target.

to qualitatively assess the behavior of the different distance functions proposed in Sec. III-C. We then present two user studies that validate our approach and further characterize the differences between alternative ways of selecting arrows within our framework.

A. Analysis of Arrow-Object Distance Functions

We first tried to characterize how different distance functions defined in Sec. III-C behave in synthetically generated scenes. We systematically varied the number of objects in the scene, the size and shape of the objects, the relative placements of the objects, and the size of the no-projection buffer around objects. Fig. 2 presents the outcome of our optimization with the five distance functions in four example test scenes. Scene1 demonstrates how d_A differs from others in that it only forces the arrow to point towards the object and does not move it closer to the object in the absence of distractors. In Scene2 we isolate the effect of the distractor. The d_A function now finds an arrow pose that is pointing at the target while pointing as much away from the distractor as possible. The other functions trade off pointing away from the distractor with being far away from the distractor. Scene3 in comparison to Scene2 demonstrates how d_D and d_E are impacted by the size of the distractor object, pointing further away from the distractor. Scene4 demonstrates the interaction between multiple distractors.

All scenes shown in Fig. 2 allow each of the distance functions to unambiguously indicate the correct target object. It is clear that an object can be unambiguously indicated by any arrow pointing towards the target, such that the ray from the arrow will first intersect with the target, if there is sufficient space around it to fit the arrow. In these situations we expect the difference between the alternative distance functions to be only subjective. In more complex scenarios with clutter and unprojectable regions around the target object, the alternative functions can have more or less ambiguity resulting in errors and delays in inferring the reference, as we will see in Sec. IV-B.

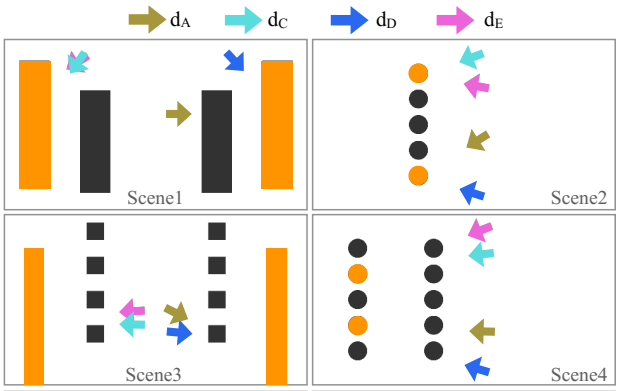

Fig. 3: Test scenes from the online evaluation. The target in each prompt was one of the two orange objects. Participants saw all objects as black and the arrow as red in all prompts.

B. Online Evaluation

We performed an online user study to validate our arrow placement model on 2D scenes and compared alternative distance functions. We tested first on 2D scenes to verify our model on simple cases, those without the nuanced perspective transformations that arise with angled 3D viewpoints. A 2D scene could also be thought of as a representation of a 3D tabletop scene from a top-down orthographic view, with the 2D image representing the table surface and shapes within the image as objects on top.

1) Study design and procedure: Our study involved two parts. The first included a series of object identification tasks in which the participant was presented with a scenario: a 2D scene composed of shapes and an arrow precomputed with our method using one of the distance functions proposed in Sec. III-C. To keep the number of compared alternatives manageable we removed d_B as it resulted in the same (or equivalent) arrows in the absence of irregularly shaped objects. For each prompt, the objects in the scene were presented first and the arrow was presented after a 5-second countdown. Participants were told to click on the object they thought the arrow was pointing at as soon as the arrow appeared. After selection, the next prompt started.

We created one practice scene and four test scenes (Fig. 3). The target object for each scene was one out of two targets that were equivalent due to the symmetry of the scene. This was done to prevent participants from using memory from previous prompts to guess the target. We conducted a within-participants study, where all participants saw all prompts corresponding to combinations of four distance functions in four scenes. Participants first had two practice rounds in the practice scene, and then saw the 16 prompts in counterbalanced order (using latin-squares).

The second part of the study sought to capture the subjective comparison of the different distance functions. Participants were shown the same scenes from the first part, but with the target object identified and all arrows rendered in different colors on the same image. Participants were asked to rate how well each arrow indicated the target object on a 5-point Likert scale.

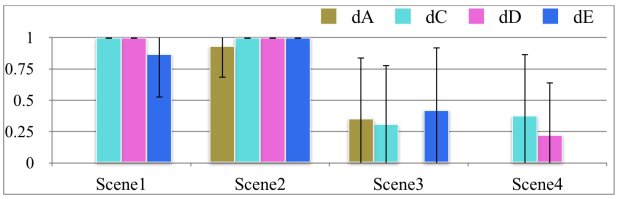

Fig. 4: Correctness of the participants' inferred target for arrows produced by four arrow-object distance functions across the four test scenes.

2) Measurements: For the object identification task, we measured the correctness of the participants' guess for the target object of the displayed arrow. We also measured the participants' response time. We expect that a hesitation in the response, indicated by longer response times, captures situations where the indication was not as clear, even if the participant eventually selected the correct answer. In the comparison task, we measured participants' subjective assessment of how well an arrow (corresponding to a certain distance function) indicated the revealed target object with a 5-point Likert scale. Since our study is an open exploration of alternative arrow-object distance functions we do not make predictive hypotheses about this dependent variable.

3) Findings: Our study was completed by 48 participants over Amazon Mechanical Turk. Fig. 4 shows the percentage of participants that correctly chose the target object based on the arrow indicator, for different scenes and distance functions. We observe that the accuracy is not 100% for all scenes and functions as we expected. Both the distance function and the scene impacted accuracy. An accuracy of 0% corresponded to cases where the arrow intersected with another object before the target object. This points to a limitation of the explored distance functions in penalizing pointing to other objects. Despite the clear ambiguities, a portion of the participants correctly identified the target with arrows produced by some functions. d_C was the only function that had non-zero accuracy in all four scenes and was at par with other functions in each scene. In an ANOVA test with distance functions and scenes as two separate factors, we found that the correctness of d_C was significantly higher than d_A and d_D ($p<0.001$) as well as d_E ($p<0.01$).

Participants took anywhere from 0.2 to 6.6 seconds to respond to prompts; however, there were no significant differences across the different distance functions. Similarly, there were no statistically significant differences between the distance functions in terms of participants' ratings of the arrows. The ratings had large variance across participants and the average ratings were around 3 on the 5-point scale for all prompts.

C. In-person Evaluation

Next, we performed an in-person study to validate our approach for 3D scenes.

1) Study design: We took the most robust distance function d_C from the online evaluation and compared it with

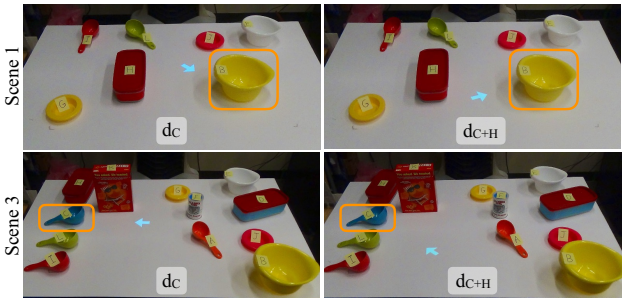

Fig. 5: Example prompts generated with d_C and d_{C+H} in three different test scenes from the lab study.

a modified version d_{C+H} that accounts for user viewpoint. This was accomplished by including the optimization term given in Eqn. 3, which pushes the arrow on the surface closer to where the user is located. The impact of the additional term can be seen in two sample scenes shown in Fig. 5.

Users stood in front of a tabletop scene, facing the PR2 robot across the table. Objects on the table were labeled with single letters (A, B, C, D, *et cetera*). There were three scenes with varying degrees of clutter and multiple prompts in each scene. Tasks were grouped by scene. For each prompt in the object identification task, participants waited for the robot to project an arrow, and spoke the label of the target object as soon as the arrow appeared. The experimenter gave a verbal cue before each prompt. Users practiced the task with one example prompt. Two dummy prompts were randomly inserted between test prompts to reduce the likelihood that users could anticipate the next prompt. The order of scenes and prompts were counterbalanced.

Each object identification task was followed by the subjective comparison task for the same scene. We asked users to compare two arrows for the same target generated using the different distance functions. Users were told the intended target and the experimenter switched between the two arrows so participants could see both. We asked them to verbally select one, both, or neither arrow as their preferred indicator and to explain their selection.

2) Measurements: For the object identification task, we measured correctness and duration from the time of the arrow's appearance to the start of the participant's verbal answer (*i.e.,* response time). For the subjective comparison task, we noted the participant's selection and transcribed the explanation. A single experimenter used ELAN [19] to annotate the recorded videos.

3) Findings: Our study was completed by 12 participants (6 female) aged 20-31 recruited from the local and university community. Fig. 6 shows the average correctness of participants' guess of the target object for the two distance functions for the three scenes. The original d_C function resulted in 83.3% overall accuracy, while the additional optimization term that accounted for user's position d_{C+H} resulted in 75.0%. However as the figure shows, this difference was mainly due to Scene 3 where there was extra clutter. The d_C function resulted in better or equivalent accuracy in all scenes, which shows that projecting into visible space is

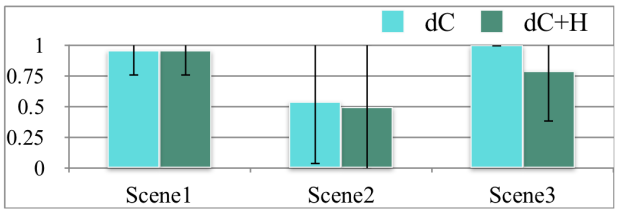

Fig. 6: Correctness of the participants' inferred target for arrows produced by d_C and d_{C+H} across three test scenes averaged across participants and prompts in the lab study.

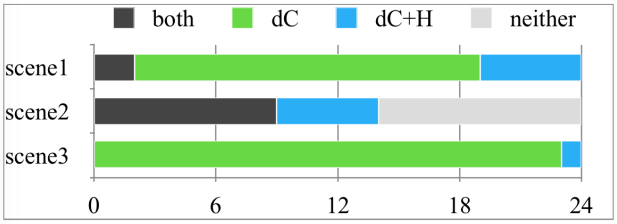

Fig. 7: Participant preferences in comparison tests across three scenes in the lab study.

sufficient and further moving the arrow towards the user is not helpful but can instead reduce accuracy. Scene 2 had the lowest accuracy due to two prompts that involved targets that had to be referenced from behind distractor objects, which were also problematic in the online study.

The two functions did not result in any difference in reaction time. Fig. 7 shows the participants preferences in the comparison prompts. In Scene 1 and 3 we see a clear preference shifted towards d_C. In Scene 2 participants liked both or neither of the prompts most of the time, but preferred d_{C+H} over d_C in some cases. Responses were mostly consistent across participants and varied more across individual prompts in a given scene. Nevertheless, our in-person study demonstrates that our system can autonomously choose arrow placements to correctly indicate a target object in most scenarios.

V. Conclusions

We present a framework for optimally choosing arrow placements for object referencing in projection-based situated human-robot communication. This framework allows for alternative arrow-object distance functions resulting in different behaviors. It generalizes to different scenes and captures a number of constraints that are relevant for making projections work on a real robot in a 3D cluttered scene with occlusions. Our evaluations demonstrate the effectiveness of the approach while pointing out limitations of the chosen distance functions in certain scenarios. We observed that some distance functions were more robust across challenging scenes, but did not result in faster response time or were not particularly preferred by participants.

Acknowledgements

This work was supported by the National Science Foundation, Awards IIS-1552427 "CAREER: End-User Programming of General-Purpose Robots" and IIS-1525251 "NRI: Rich Task Perception for Programming by Demonstration."

REFERENCES

[1] R. S. Andersen, O. Madsen, T. B. Moeslund, and H. B. Amor, "Projecting robot intentions into human environments," in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*. IEEE, 2016, pp. 294–301.

[2] R. T. Chadalavada, H. Andreasson, R. Krug, and A. J. Lilienthal, "That's on my mind! robot to human intention communication through on-board projection on shared floor space," in *2015 European Conference on Mobile Robots (ECMR)*, Sept. 2015, pp. 1–6.

[3] H. Nguyen, A. Jain, C. Anderson, and C. C. Kemp, "A clickable world: Behavior selection through pointing and context for mobile manipulation," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. 2008, pp. 787–793.

[4] D. Lazewatsky and W. Smart, "Context-sensitive in-the-world interfaces for mobile manipulation robots," in *IEEE Intl. Symp. on Robot Human Communication (ROMAN)*, 2012, pp. 989–994.

[5] S. Tellex, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," in *Proceedings of the Robotics: Science and Systems Conference*. Citeseer, 2014.

[6] J. Y. Chai, L. She, R. Fang, S. Ottarson, C. Littley, C. Liu, and K. Hanson, "Collaborative Effort Towards Common Ground in Situated Human-robot Dialogue," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '14. New York, NY, USA: ACM, 2014, pp. 33–40.

[7] R. M. Holladay, A. D. Dragan, and S. S. Srinivasa, "Legible robot pointing," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Aug. 2014, pp. 217–223.

[8] M. Salem, S. Kopp, I. Wachsmuth, K. Rohlfing, and F. Joublin, "Generation and evaluation of communicative robot gesture," *International Journal of Social Robotics*, vol. 4, no. 2, pp. 201–217, 2012.

[9] A. Sauppé and B. Mutlu, "Robot deictics: How gesture and context shape referential communication," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM, 2014, pp. 342–349.

[10] B. Mutlu, J. Forlizzi, and J. Hodgins, "A storytelling robot: Modeling and evaluation of human-like gaze behavior," in *Humanoid robots, 2006 6th IEEE-RAS international conference on*. Citeseer, 2006, pp. 518–523.

[11] R. Ros, S. Lemaignan, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken, "Which one? grounding the referent based on efficient human-robot interaction," in *19th International Symposium in Robot and Human Interactive Communication*. IEEE, 2010, pp. 570–575.

[12] H. Admoni, T. Weng, and B. Scassellati, "Modeling communicative behaviors for object references in human-robot interaction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3352–3359.

[13] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2422–2427.

[14] A. D. Dragan, K. C. T. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2013, pp. 301–308.

[15] M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in optimization and numerical analysis*. Springer, 1994, pp. 51–67.

[16] S. Johnson, "The nlopt nonlinear-optimization package [software]," 2014.

[17] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[18] G. Bradski and A. Kaehler, "Opencv," *Dr. Dobbs journal of software tools*, vol. 3, 2000.

[19] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "Elan: a professional framework for multimodality research," in *5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006, pp. 1556–1559.