Projecting Robot Navigation Paths: Hardware and Software for Projected AR

Zhao Han,*‡, Jenna Parrillo†, Alexander Wilkinson†, Holly A. Yanco† and Tom Williams*
*Department of Computer Science, Colorado School of Mines, Golden, Colorado, USA 80401
Email: zhaohan@mines.edu, twilliams@mines.edu
†Department of Computer Science, University of Massachusetts Lowell, Lowell, Massachusetts, USA 01854
Email: jenna_parrillo@student.uml.edu, alexander_wilkinson@student.uml.edu, holly@cs.uml.edu

***Abstract*—For mobile robots, mobile manipulators, and autonomous vehicles to safely navigate around populous places such as streets and warehouses, human observers must be able to understand their navigation intent. One way to enable such understanding is by visualizing this intent through projections onto the surrounding environment. But despite the demonstrated effectiveness of such projections, no open codebase with an integrated hardware setup exists. In this work, we detail the empirical evidence for the effectiveness of such directional projections, and share a robot-agnostic implementation of such projections, coded in C++ using the widely-used Robot Operating System (ROS) and rviz. Additionally, we demonstrate a hardware configuration for deploying this software, using a Fetch robot, and briefly summarize a full-scale user study that motivates this configuration. The code, configuration files (roslaunch and rviz files), and documentation are freely available on GitHub at *https://github.com/umhan35/arrow_projection*.**

***Index Terms*—Robot navigation, navigation intent, projected augmented reality (AR), ROS, rviz, open science**

I. Introduction

Robots are increasingly navigating around populous areas and moving along with people. This is true for service robots found in airports, hotels, restaurants, delivery robots on sidewalks, mobile robots in warehouses, and autonomous cars. For these robots to be deployed safely, humans around them must understand their navigation intent. HRI researchers have traditionally focused on arm movement intent [2], [3], exploring eye gaze [4], [5] and other non-verbal means [6] for stationary robots to better convey such intent. Recently, however, HRI researchers have begun to explore the externalization or visualization of robotic navigation intent (e.g. path plans) as well [7], [8], [9], [10], [11], [12], to enable more understandable robot's navigation behaviors, so as to improve trust and acceptance.

One method for conveying navigation intent is through directional projections [7], [9], [10], [11] such as arrow projections [7], [11], where the robot is equipped with a projector and projects directional cues indicating which direction the robot intends to move. These cues can take the forms of lines [9], gradient bands [10], or arrows [7], [11]. Compared to head-mounted see-through augmented reality [15], [16], projector-based augmented reality does not require interactants to wear special hardware, allowing visualizations to be seen by many observers at once, facilitating usability in group, team, or crowd contexts [17], [18], [19]. As we will discuss in

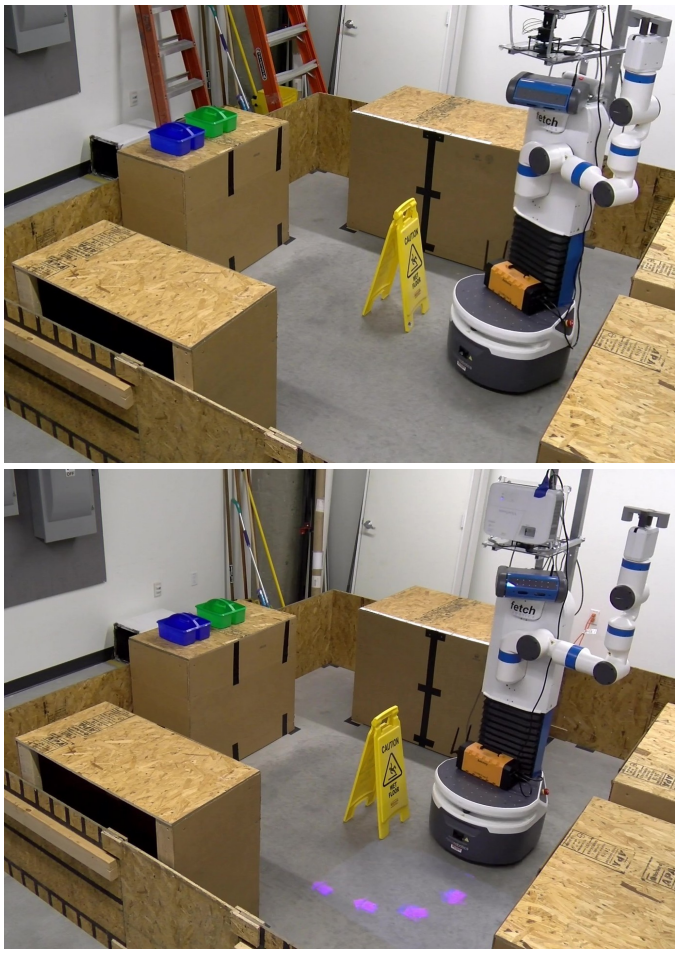

Fig. 1. A Fetch mobile manipulator navigating around a wet floor sign without (upper) and with (lower) arrow projection (purple). Despite of the effectiveness of arrow projection for revealing navigation path, no public code is readily available. Here we share such code in ROS and provide hardware support. Additional **videos** are available in [13], [14].

‡Most of this work was completed while Zhao Han was affiliated with the University of Massachusetts Lowell. This work has been supported in part by the Office of Naval Research (N00014-18-1-2503) and the National Science Foundation (IIS-1909864). We also thank Brian Flynn at the University of Massachusetts Lowell NERVE Center [1] for building the projector support with mechanical springs.

Section II, a number of human-subjects studies [7], [9], [10], [11] have shown the effectiveness of this approach [10], [11].

In this work, we share a practical solution for robotic arrow projections, describe the nature of the implementation of this solution, and provide access to the code for that solution. Additionally, we provide a specific robot and off-the-shelf projector that we have successfully used, as well as a summary of a full-scaled user study to provide more empirical evidence. For the implementation, we used the popular Robot Operating System (ROS) [20] and its visualization tool, rviz [21], for rendering the arrows. Using ROS made the implementation robot-agnostic, as most robots used in research and development have ROS support [22]. Even non-ROS frameworks [23], [24], and cognitive architectures, such as DIARC [25], typically include bridges to ROS. Using rviz as the rendering engine eliminates the need for practitioners and researchers to learn a tool outside of the ROS ecosystem, such as Unity. Moreover, rviz has a GUI and does not require any computer graphics programming.

Given the wide range of applications of arrow projection on different robots and its proven effectiveness and efficiency, we believe our work is highly relevant and beneficial to the HRI community. The readily available implementation with step-by-step documentation also allows researchers to focus on other interesting and emerging issues under social navigation, e.g., human navigation modeling [26], [27], [28], [29] and robot navigation that obeys social or moral norms [30], [31]. We welcome GitHub issues for questions and pull requests.

II. Related Work on The Effectiveness and Improved Perception of Direction Projection

There is significant empirical evidence that directional projections have numerous quantifiable benefits. However, these previous works have not provided publicly accessible software implementations. Originally proposed to solve the accessibility issue caused by touch screen placement height, Park and Kim [32]'s work was among the first attempts to equip a projector onto a robot, projecting a graphical user interface (GUI) onto the floor so people at different height can see the interface.

Before using arrows, other directional projections were proposed. Inspired by Park and Kim [32], Chadalavada et al. [9] proposed projecting a line with a grid onto the ground to indicate navigation path and collision avoidance range of a robot. Even with this primitive geometry element, Chadalavada et al. [9]'s work suggests potential for enhanced participant perception of the robot across multiple attributes when projections were used, including communication, reliability, predictability, transparency, and situation awareness. Although no statistical tests were ran, their descriptive statistics provide one of the first pieces of evidence for directional projection effectiveness.

Instead of lines, Watanabe et al. proposed projecting a gradient light band in a hallway to show the navigation trajectory to be followed by a robotic wheelchair with a wheelchair user [10]. The projection was compared with no projection, and with a projection paired with a screen. Their results suggested that adding projections enhanced comfortability and perceived motion intelligence.

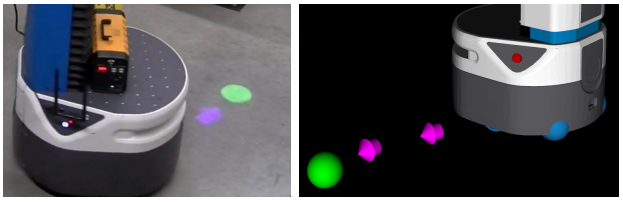

Fig. 2. **Left**: A destination circle in green to indicate the end of a navigation path. **Right**: Arrows with a destination circle rendered in rviz.

Furthermore, Watanabe et al. [10] found that when projections were used, nearby walkers chose not to stop and stepped away towards walls earlier. These behavioral changes are further evidenced by Chadalavada et al. [11], who show that projection onto a shared floor space encourages participants to choose an alternative safer path.

Finally, Coovert et al. used arrow projection to show a robot's path and investigated navigation intention prediction during different periods while the robot was avoiding traffic cones in a hallway [7]. Their work showed that adding projections led to observable differences at a variety of time scales, as participants were typically able to quickly identify the robot's navigation intent. Moreover, participants also displayed high confidence in their identifications.

Researchers have also begun to recently compare between some of these previously presented visualizations. Chadalavada et al. [11], for example, recently compared line and arrow visualizations, and found that arrows are typically preferred compared to lines and blinking arrows. In our own work, we thus made an informed decision to use arrow projections to visualize robot navigation intent. We also added a solid circle to indicate navigation destination (Fig. 2 left).

III. Hardware Setup: Robot, Projector, Power

Before we detail the software, we first describe a hardware configuration in which we validated our software. In this configuration , a projector is mounted onto a Fetch robot [33], as shown in Fig. 1. Fetch is a mobile manipulator robot with a single 7-DOF arm mounted onto its chest. Its height ranges from $1.096m - 1.491m$ $(3.596ft - 4.892ft)$. The projector we used is ViewSonic PA503W [34], as shown on the top of the images in Fig. 1. It used the DLP technology in its lamp, has a brightness of 3,800 ANSI lumens and 22,000:1 contrast ratio, making sure the projection is still legible in indoor environments with lights on. DLP has higher brightness than LCDs because LCDs are transmissive and the heat generated cannot be easily dispatched [35]. We chose this projector as our prototype, but other projectors will also work as long as the purchaser makes sure to consider these same three factors, i.e., the lamp technology, ANSI lumens and contrast ratio.

Although we chose the specific robot and projector, our implementation is robot- and projector-agnostic. The only requirement in terms of hardware is the mounting point of a

projector must be within the projector's throw distance range to make sure the projection is not blurred. The range should be available on the product specs page [34].

Because we planned to pan and tilt the projector for tabletop projection, we attached a ScorpionX MX-64 Robot Turret [36] to actuate the projector. This is not necessary if only used for ground projection, for which the turret is constantly tilted down to the maximum angle. Instead, the projector can be mounted at a fixed position. However, if the actuation of the projector is needed, we do recommend stabilizing the projector, in our case, by using mechanical springs, as shown in Fig. 1 top.

Another hardware element that might be needed is a portable power station to power the projector as the robot will be untethered. For ViewSonic PA503W, it needs AC power ($100-240V \pm 10\%, 50/60Hz$) and has a typical consumption of $260W$, which we found most portable power bank cannot provide (e.g., [37]), leading to projector powered off soon after turned on. We did find a working portable power station with $500W$ AC power outlet [38]. However, if another projector is chosen under the aforementioned three considerations with less power consumption, the power bank is no longer needed.

IV. Implementation in ROS

Once a projector is mounted onto a robot and pointing towards the front of a robot's base, the projector's pose relative to the robot needs to be integrated into the robot's transform hierarchy using the "static transform publisher" node [39] from the "tf" package [40]. A sample ROS launch file [41] is provided in "tf_publisher.launch".

Note that to make our account of the implementation beginner-friendly, we added relevant ROS learning resources to references throughout this and following section.

To retrieve the global navigation path, we subscribed to the ROS topic "/move_base/TrajectoryPlannerROS/global_plan" exposed by the ROS navigation stack [42], [43]. The topic encloses a Path message [44] from the nav_msgs package [45], including poses discretizing the continuous navigation path.

However, our approach uses a voxelized map, an adaptive Monte Carlo localization [46], and an A*-based path planner [47], and as such, the poses in the path are not evenly spaced. Because an arrow has a physical size, the arrows may overlap and thus obscure each other if the distance between a pair of poses is too close. We thus created Algorithm 1 to solve this problem. The algorithm also allows specifying a navigation destination circle, as shown in Fig. 2 left.

Algorithm 1 iterates each point from the destination to the starting point (Line 1, 2, and 13). The destination point is always the first point in P' (Line 3) because we wanted the destination circle bigger than an arrow. For each point p in the path, it will skip to the point p' when the distance between p and p' is bigger than the desired distance D (Line 9 before "or"). We stop iterating once the index becomes negative and is accessed (Line 7), caught by the try-catch block (Line 4, 10, and 11), or the distance is shorter than D in the beginning (Line 9 after "or"). The algorithm is implemented in C++ and available in the get_sparse_points function in "PathProjection.cpp".

Algorithm 1: Evenly Space Out ROS Nav. Path Points

Input: ROS Global Path Poses P *// Unevenly spaced*
Input: Double D *// Distance between arrows*
Input: Double $\varnothing$ *// Destination circle diameter*
Output: Array[x,y,z] P'

1 $i \leftarrow |P| - 1$ *// From destination to starting point*
2 **repeat**
3 $\quad p \leftarrow P[i]$, $P' \leftarrow P' \cup \{p\}$, $i' \leftarrow i$
4 $\quad$ **try**
5 $\quad\quad$ **repeat**
6 $\quad\quad\quad i' \leftarrow i' - 1$
7 $\quad\quad\quad p' \leftarrow P[i'].pose.position$ *// ROS quirk*
8 $\quad\quad\quad d \leftarrow \sqrt{(p.x - p'.x)^2 + (p.y - p'.y)^2}$
9 $\quad\quad$ **until** $d < D$ ***or*** $(i = |P| - 1$ ***and*** $d < D + \varnothing)$
10 $\quad$ **catch** *Array Out of Bound Exception*
11 $\quad\quad$ *// Done.* $i' < 0$ *now. Line 13 breaks the loop.*
12 $\quad i \leftarrow i'$
13 **until** $i > 0$
14 **return** P' *// Evenly spaced*

All the logic in this section is coded in the PathProjection ROS node [48] in "path_projection_node.cpp", "PathProjection.h", and "PathProjection.cpp".

A. Visualizing Arrows in rviz

Once we get a list of evenly spaced-out points from the ROS navigation path, we create an array of markers [49] as a MarkerArray message [50] in the "visualization_msgs" ROS package [51], and publish it so it can be added to rviz [52].

In rviz, an arrow consists of a shaft and head. Under the hood, as seen from its arrow_marker.cpp [53] and arrow.cpp [54] source code, rviz uses four parameters for an arrow – shaft length, shaft diameter, head length, and head diameter.

However, the rviz GUI only allows for specifying three parameters, the scale 3D vector through the Marker message [49] interface, partly because the message was designed to be generic to specify other shapes such as cubes, spheres, or cylinders [49]. The generic design turned out to limit customization of the arrow. To tackle this problem, we spent significant time investigating rviz's source code and directly modified line 108 of arrow_marker.cpp [53], in which the four parameters of the arrow's shaft and head are set, in . Specifically, we made the shaft's length the same as the arrow's head's. A diff file is available in "arrow_marker.cpp.diff". We hope this can help roboticists to spend more time on their research of interest instead of on arrow customization.

B. Projecting Arrows in rviz via Projector Lens

Once the arrow list is published, we can subscribe to the arrow list's ROS topic in order to render the arrows in rviz (Fig. 2 right). This renders each arrow at position, $(x, y, 0)$, where 0 denotes rendering on the ground plane. In theory,

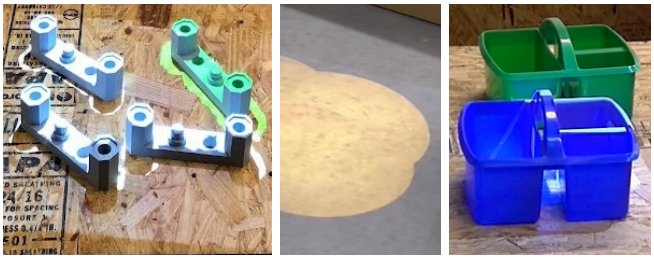

Fig. 3. Three other visualizations to show the generalizability for our work: point cloud for perceived objects [58], multiple spheres for ground obstacle, cube for caddy section. Any rviz visualization can be projected.

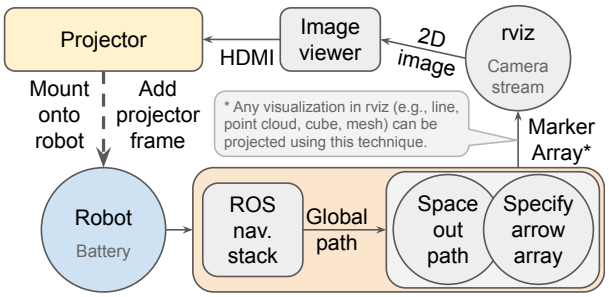

Fig. 4. High-level representation of the arrow projection implementation, with a note on the generalizability of our technique to other forms of visualizations.

if we can place a virtual camera in rivz at the pose relative to the robot, exactly the same as where the projector is mounted towards the ground (See Fig. 1), we can get a 2D image from the camera and it becomes the input to the projector. This is accomplished using the "rviz camera stream" plugin [55]. We provide a sample rviz configuration file in "path_projection.rviz". Then we used the image view node [56] to subscribe to that image topic, and made it full screen in Ubuntu's keyboard settings (Toggle full-screen mode).

Finally, to make the projection to the physical world not distorted, the projector lens needs to be calibrated, which we used the pinhole lens model, as seen in [57], [58]. A sample CameraInfo message [59] and a launch file are provided in "projector_camera_info.yaml" and "camera_publisher.launch".

V. Evaluation

In this section, we describe the experimental validation of our arrow projections in the context of human-robot communication during a collaborative mobile manipulation task [60], where a Fetch robot projected arrows to show its detour path, spheres to indicate obstacles, and projected point clouds [58] to indicate mis-/recognized objects onto a table (cp. the similar visualization strategy of Reardon et al. [61] in field navigation contexts). In our approach, ground and tabletop projections were compared with the following conditions: physical replay of its past navigation path and manipulation plans (pick and place), and/or speech describing the locations of (a) the objects robot picked/placed and (b) the obstacles around which robots detoured.

This experiment assessed whether our visualizations enabled participants who were temporarily not collocated with the robot to infer the identities and locations of obstacles and of misrecognized or misplaced objects, and the reason for detours. Arrow projections, specifically when paired with sphere projections enabled 93.7% accuracy for inferring the location of obstacles requiring detours. In contrast, speech-based communication only afforded 83.2% accuracy.

As such, this experiment provided empirical evidence for the benefits of our approach. Moreover, we again stress that using our approach, any visualizations that can be added to rviz can be projected (Fig. 3), as long as they are in the virtual camera's view. This includes basic shapes, such as cubes, spheres, and cylinders (visualization_msgs/Marker [49]) as well as complex objects specified in polygon mesh (MESH_RESOURCE in Marker [49]; See [62] for its usage). However, there are a number of remaining limitations to this approach.

VI. Limitations

Because the projector must be mounted at a fixed position, it introduces a few limitations. First, the projector must be mounted above the lower value of the projection throw distance range, defined as the distance between projector lens and projection surface. Otherwise, projection onto the ground becomes blurred. This is not a problem for a tall robot, such as the Fetch mobile manipulator, whose height ranges $1.096m - 1.491m$ $(3.596ft - 4.892ft)$, within the throw distance of ViewSonic PA503W $0.99m - 10.98m$ $(3.25ft - 36.02ft)$. However, mobile robots without upper bodies are often lower than $1m$, such as TurtleBot or PyRobot [63]. Using such robots thus requires building a tall structure on the robot, or selecting a projector with a short throw distance.

Another limitation is projection size. This can be clearly seen from Fig. 1 bottom, the slightly brighter area, which is limited to around $55 \times 98cm^2$ in a $16:9$ ratio. This results in a small projection area, preventing projecting arrows farther along a navigation path, which may lead to problems in large spaces like warehouses. One solution is to mount another projector lower, pointing the ground further away, similar to the headlight of a car, so that the projection can be thrown further to cover a long area, although this may artificially inflate arrow size. Fortunately, because we know the distance between arrows and the projector lens, we can shrink the size of arrows proportionally to distance.

VII. Conclusions

We provided arrow projection software and summarized the demonstrated benefits of this software. As shown in Fig. 4, to project arrows, one needs to mount a projector onto a robot with a portable power station, integrate the projector's pose into the robot's transform hierarchy, space out the ROS navigation stack's global path, render the associated arrows in rviz, and finally output to the calibrated projector. Additionally, we also showed the generalization of our technique, i.e., the capability to project any rviz visualizations.

REFERENCES

[1] "The New England Robotics Validation and Experimentation (NERVE) Center at the University of Massachusetts Lowell," https://www.uml.edu/Research/NERVE/.

[2] M. J. Gielniak and A. L. Thomaz, "Generating anticipation in robot motion," in *2011 RO-MAN*. IEEE, 2011, pp. 449–454.

[3] M. Kwon, S. H. Huang, and A. D. Dragan, "Expressing robot incapability," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 87–95.

[4] A. Moon, D. M. Troniak, B. Gleeson, M. K. Pan, M. Zheng, B. A. Blumer, K. MacLean, and E. A. Croft, "Meet me where i'm gazing: how shared attention gaze affects human-robot handover timing," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, 2014, pp. 334–341.

[5] H. Admoni and B. Scassellati, "Social eye gaze in human-robot interaction: a review," *Journal of Human-Robot Interaction*, vol. 6, no. 1, pp. 25–63, 2017.

[6] S. Saunderson and G. Nejat, "How robots influence humans: A survey of nonverbal communication in social human–robot interaction," *International Journal of Social Robotics*, vol. 11, no. 4, pp. 575–608, 2019.

[7] M. D. Coovert, T. Lee, I. Shindev, and Y. Sun, "Spatial augmented reality as a method for a mobile robot to communicate intended movement," *Computers in Human Behavior*, vol. 34, pp. 241–248, 2014.

[8] D. Szafir, B. Mutlu, and T. Fong, "Communicating directionality in flying robots," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2015, pp. 19–26.

[9] 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)*, 2015, pp. 1–6.

[10] A. Watanabe, T. Ikeda, Y. Morales, K. Shinozawa, T. Miyashita, and N. Hagita, "Communicating robotic navigational intentions," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5763–5769.

[11] R. T. Chadalavada, H. Andreasson, M. Schindler, R. Palm, and A. J. Lilienthal, "Bi-directional navigation intent communication using spatial augmented reality and eye-tracking glasses for improved safety in human–robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101830, 2020.

[12] T. Blenk and S. Cramer, "Lane change decision making for automated driving," in *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 6–15.

[13] "Accompanying video 1, with destination circle," https://osf.io/3d8c5/?view_only=09bc6b19cb144b8e8c689f00eb59e828.

[14] "Accompanying video 2," https://osf.io/u2g5x/?view_only=09bc6b19cb144b8e8c689f00eb59e828.

[15] T. Williams, D. Szafir, T. Chakraborti, and H. Ben Amor, "Virtual, augmented, and mixed reality for human-robot interaction," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 403–404.

[16] T. Williams, N. Tran, J. Rands, and N. T. Dantam, "Augmented, mixed, and virtual reality enabling of robot deixis," in *International Conference on Virtual, Augmented and Mixed Reality*. Springer, 2018, pp. 257–275.

[17] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[18] S. Sebo, B. Stoll, B. Scassellati, and M. F. Jung, "Robots in groups and teams: a literature review," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–36, 2020.

[19] R. Oliveira, P. Arriaga, and A. Paiva, "Human-robot interaction in groups: Methodological and research practices," *Multimodal Technologies and Interaction*, vol. 5, no. 10, p. 59, 2021.

[20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, 2009.

[21] D. Gossow, A. Leeper, D. Hershberger, and M. Ciocarlie, "Interactive markers: 3-d user interfaces for ros applications [ros topics]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 14–15, 2011.

[22] "Official ROS robot showcase," https://robots.ros.org/.

[23] J. Kramer and M. Scheutz, "Development environments for autonomous mobile robots: A survey," *Autonomous Robots*, vol. 22, no. 2, pp. 101–132, 2007.

[24] A. Elkady and T. Sobh, "Robotics middleware: A comprehensive literature survey and attribute-based bibliography," *Journal of Robotics*, 2012.

[25] M. Scheutz, T. Williams, E. Krause, B. Oosterveld, V. Sarathy, and T. Frasca, "An overview of the distributed integrated cognition affect and reflection diarc architecture," *Cognitive architectures*, pp. 165–193, 2019.

[26] S. J. Guy, M. C. Lin, D. Manocha *et al.*, "Modeling collision avoidance behavior for virtual humans." in *AAMAS*, vol. 2010, 2010, pp. 575–582.

[27] P. Ratsamee, Y. Mae, K. Ohara, M. Kojima, and T. Arai, "Social navigation model based on human intention analysis using face orientation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1682–1687.

[28] G. Ferrer and A. Sanfeliu, "Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1730–1735.

[29] Y. Che, C. T. Sun, and A. M. Okamura, "Avoiding human-robot collisions using haptic communication," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5828–5834.

[30] S. B. Banisetty and T. Williams, "Implicit communication through social distancing: Can social navigation communicate social norms?" in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 499–504.

[31] S. B. Banisetty, S. Forer, L. Yliniemi, M. Nicolescu, and D. Feil-Seifer, "Socially aware navigation: A non-linear multi-objective optimization approach," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, no. 2, pp. 1–26, 2021.

[32] J. Park and G. J. Kim, "Robots with projectors: an alternative to anthropomorphic hri," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, 2009, pp. 221–222.

[33] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich, "Fetch and freight: Standard platforms for service robot applications," in *Workshop on autonomous mobile service robots*, 2016, fetch robot webpage: https://fetchrobotics.com/fetch-mobile-manipulator/.

[34] "ViewSonic PA503W," https://www.viewsonic.com/us/pa503w.html.

[35] L. J. Hornbeck, "Digital light processing for high-brightness high-resolution applications," in *Projection Displays III*, vol. 3013. International Society for Optics and Photonics, 1997, pp. 27–40.

[36] "ScorpionX MX-64 Robot Turret," https://www.trossenrobotics.com/p/ScorpionX-RX-64-robot-turret.aspx.

[37] "Jackery Portable Power Station," https://www.amazon.com/dp/B07D29QNMJ.

[38] "Aeiusny Portable Solar Generator," https://www.amazon.com/dp/B095P7QX3G/.

[39] "The static_transform_publisher ROS node," https://wiki.ros.org/tf#static_transform_publisher.

[40] "The tf ROS package," https://wiki.ros.org/tf.

[41] "The roslaunch tool," https://wiki.ros.org/roslaunch.

[42] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 300–307.

[43] "The navigation stack in ROS," https://wiki.ros.org/navigation.

[44] "The Path message from the nav_msgs ROS package," https://wiki.ros.org/nav_msgs.

[45] "The nav_msgs ROS package," https://wiki.ros.org/nav_msgs.

[46] D. Fox, "Kld-sampling: Adaptive particle filters and mobile robot localization," *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[47] K. Konolige, "A gradient method for realtime robot control," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, vol. 1. IEEE, 2000, pp. 639–646.

[48] "Understanding Nodes in the official ROS tutorial," https://wiki.ros.org/ROS/Tutorials/UnderstandingNodes.

[49] "The generic Marker message in the visualization_msgs ROS package," https://docs.ros.org/en/api/visualization_msgs/html/msg/Marker.html.

[50] "The MarkerArray message in the visualization_msgs ROS package," https://docs.ros.org/en/api/visualization_msgs/html/msg/MarkerArray.html.

[51] "The visualization_msgs ROS package," https://ros.org/wiki/visualization_msgs.

[52] "rviz: The 3D visualization tool for ROS," https:///wiki.ros.org/rviz.

[53] "arrow_marker.cpp file in the rviz source code," https://github.com/ros-visualization/rviz/blob/melodic-devel/src/rviz/default_plugin/markers/arrow_marker.cpp.
[54] "arrow.cpp file in the rviz source code," https://github.com/ros-visualization/rviz/blob/melodic-devel/src/rviz/ogre_helpers/arrow.cpp.
[55] "The rviz camera stream ROS package – a rviz plugin," https://github.com/uml-robotics/rviz_camera_stream.
[56] "image_view: A simple viewer for ROS image topics." https:///wiki.ros.org/image_view.
[57] D. Wang, C. Kohler, A. ten Pas, A. Wilkinson, M. Liu, H. Yanco, and R. Platt, "Towards assistive robotic pick and place in open world environments," in *Proceedings of the International Symposium on Robotics Research (ISRR)*, 2019.
[58] Z. Han, A. Wilkinson, J. Parrillo, J. Allspaw, and H. A. Yanco, "Projection mapping implementation: Enabling direct externalization of perception results and action intent to improve robot explainability," *the AI-HRI Symposium at AAAI-FSS 2020*, 2020.
[59] "The CameraInfo message in sensor_msg ROS package." https://docs.ros.org/en/api/sensor_msgs/html/msg/CameraInfo.html.
[60] Z. Han and H. A. Yanco, "Explaining a robot's past: Communicating missing causal information by replay, speech and projection," *ACM Transactions on Human-Robot Interaction (THRI)*, Under review.
[61] C. Reardon, K. Lee, and J. Fink, "Come see this! augmented reality to enable human-robot cooperative search," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2018, pp. 1–7.
[62] "Ros wiki section for mesh resource rviz marker," https://wiki.ros.org/rviz/DisplayTypes/Marker#Mesh_Resource_.28MESH_RESOURCE.3D10.29_.5B1.1.2B-.5D.
[63] A. Murali, T. Chen, K. V. Alwala, D. Gandhi, L. Pinto, S. Gupta, and A. Gupta, "Pyrobot: An open-source robotics framework for research and benchmarking," *arXiv preprint arXiv:1906.08236*, 2019.