

Projection Mapping Implementation: Enabling Direct Externalization of Perception Results and Action Intent to Improve Robot Explainability

Zhao Han[†], Alexander Wilkinson, Jenna Parrillo, Jordan Allspaw, Holly A. Yanco

Department of Computer Science
University of Massachusetts Lowell
One University Ave, Lowell, MA, USA

Abstract

Existing research on non-verbal cues, e.g., eye gaze or arm movement, may not accurately present a robot’s internal states such as perception results and action intent. Projecting the states directly onto a robot’s operating environment has the advantages of being direct, accurate, and more salient, eliminating mental inference about the robot’s intention. However, there is a lack of tools for projection mapping in robotics, compared to established motion planning libraries (e.g., MoveIt). In this paper, we detail the implementation of projection mapping to enable researchers and practitioners to push the boundaries for better interaction between robots and humans. We also provide practical documentation and code for a sample manipulation projection mapping on GitHub: github.com/uml-robotics/projection_mapping.

1 Introduction

As robots are increasingly deployed in areas ranging from factories and warehouses to hotels and private homes, there has been a growing interest in having robots explain their behaviors and actions. Past research has shown that improving the understanding of robots improves trust in real-time (Desai et al. 2013) and leads to greater efficiency during more difficult human-robot collaboration scenarios (Admoni et al. 2016).

Traditionally, because of the distinct embodiment feature of physical robots, human-robot interaction (HRI) researchers have been focused on how to enable robots to communicate their intention through non-verbal means that are typical among humans, most notably pointing through eye gaze (Moon et al. 2014; Admoni and Scassellati 2017), and arm movement (Dragan, Lee, and Srinivasa 2013; Kwon, Huang, and Dragan 2018). Light as an indication has also been used (Szafir, Mutlu, and Fong 2015).

These non-verbal cues found in human life play an important role in supporting and improving communication (Admoni and Scassellati 2015), but can also cause confusion. For example, how can a robot communicate which objects it has detected and which one is it going to grasp? In cases

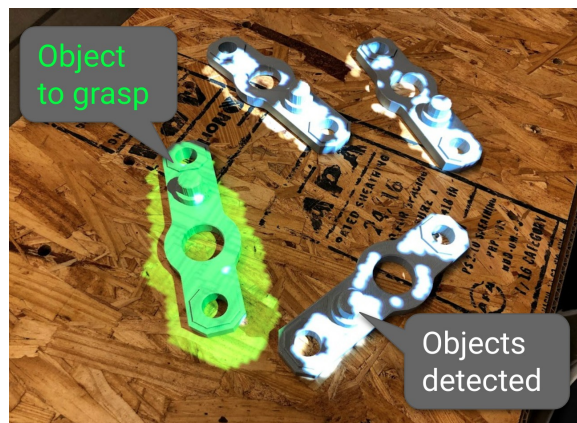


Figure 1: The projection of perception results: the detected objects (white and green) and the object to be manipulated (green). Using our implementation of projection mapping, researchers and practitioners can enable a robot to accurately externalize internal states for explanation. A video is available at youtu.be/S0z9e2gUrEA.

like these, using eye-gaze and/or pointing with its arm or end effector can be vague, especially in a clustered environment (e.g., for four clustered objects on a table). Even with verbal explanations, these gestures could be underspecified, requiring follow-up questions for clarification.

In this paper, we present a tool for implementing projection mapping using an off-the-shelf projector, including the high level architecture and low level technical details, in order to project perception results directly onto non-flat objects of interest in the environment. We also describe a concrete robot and hardware platform as an example of the tool’s use, even though the technique is robot-agnostic. Note that while we focus on projection mapping with the same object as input and output, projection mapping can be applied more broadly to arbitrary objects and targets (Grundhöfer and Iwai 2018).

While projection mapping in robotics is not a new idea, the implementation effort has been missing and thus not as accessible as implementing arm movement or eye gaze

[†]Corresponding author: zhan@cs.uml.edu
Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

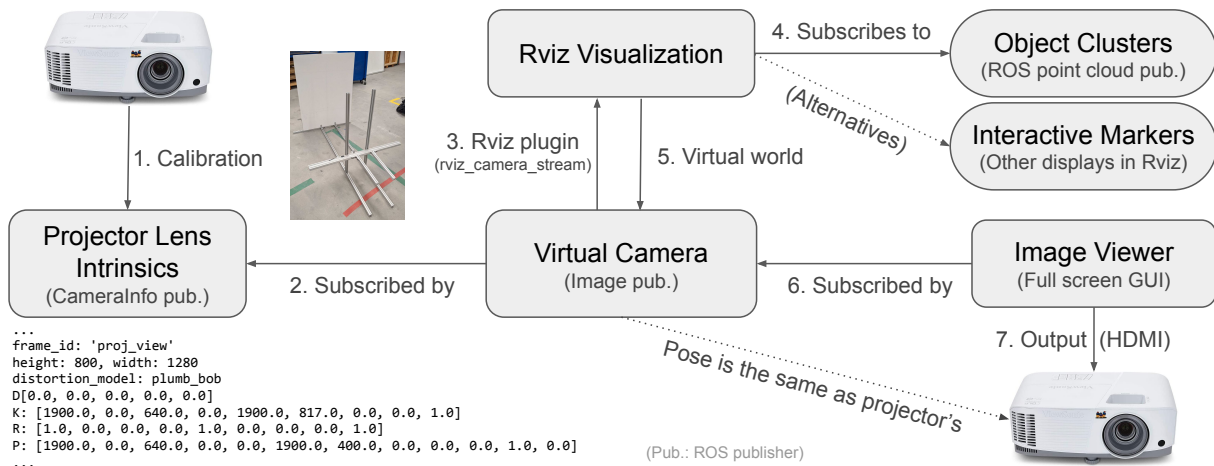


Figure 2: High-level diagram for our projection mapping implementation. With the projector lens calibrated, a virtual camera – placed in Rviz with the same pose as the projector in real world – subscribes to the camera intrinsics so it can output an image of objects visualized in the virtual world in Rviz to the projector to reflect the perceived objects. See Section 3 for more details.

through head movement. This gap effectively blocks HRI researchers from conducting human-subject studies to investigate the effects of accurate externalization through projection mapping or by comparing it to other methods. In addition, this work can also help robotics and AI researchers to externalize the output of their computer vision algorithms for a better understanding of their algorithms.

As opposed to the non-verbal methods and verbal explanations found among humans, projection mapping is a method that allows for *direct and accurate* externalization. This projection completely removes the need for mental inference as the perceived objects or the objects to be manipulated are directly externalized. The directness is similar to the use of a display screen, but projection is more *salient* because bystanders or robot coworkers can also see the projection from farther away, instead of requiring people to stop their work in order to walk to a monitor to examine the robot’s states. Direct projection onto the operating environment also *eliminates mental mapping* from another media such as a monitor, which can cause misjudgment and lead to undesired consequences.

2 Related Work

Compared to typical use cases of a projector, such as watching videos or presenting slides, the usage of a projector in the robotics field is rather rare. A more popular approach has been leveraging virtual reality headsets (e.g., (Allspaw et al. 2018; Chakraborti et al. 2018; Rosen et al. 2020)), which are readily available commercially, including devices such as the Oculus Rift and the HTC Vive along with their Software Development Kits. Some researchers have explored use cases of projection mapping (e.g., (Andersen et al. 2016)) or simple projection methods (e.g., (Chadalavada et al. 2015; Watanabe et al. 2015; Covert et al. 2014; Ghiringhelli et al. 2014)) that require no extra headset to be worn by the users; however, as previously stated, the lack of readily available implementation has impeded further advancement.

Andersen et al. used a projector to reveal a robot’s intent and task information onto the workspace for better human-robot collaboration (2016). Specifically, the robot constantly projects the wireframe of a car door to help a factory worker to understand the robot’s perception accuracy; additionally, before part manipulation, the robot will project the segment of interest to inform the worker of its next manipulation target. An experiment was conducted to compare projection with display screens for a cube moving and rotating task. Results show that there were fewer performance errors and fewer questions asked with projection.

Recently, Gao and Huang proposed (Gao and Huang 2019) using a projector to project an interface on a flat tabletop surface for robot programming. However, the focus is on the ease of robot programming, and the projection does not require mapping objects back to their corresponding real-world objects. Similarly, many researchers have been using projection to project onto another flat surface, the floor, to indicate navigation intent. Chadalavada et al. used lines to indicate the path plan and the collision avoidance range (2015). Watanabe et al. used a band of light to externalize the path (2015) while Covert et al. used arrows for the path (2014).

In our work, one of the implementation efforts was mapping perceived objects (e.g., from a depth camera) back to another view point (projector) with the same distortion caused by the view point difference, rather than flat surface projection. Nonetheless, we also provide a navigation path projection using our method in the aforementioned GitHub repository.

3 Projection Mapping Implementation

Figure 2 shows the high-level architecture of our implementation. Essentially, we set up a virtual camera with the same lens intrinsics in the same pose as the projector’s in the physical world in the ROS visualization software, Rviz. We then publish perception results in point cloud clusters and the ob-

ject point cloud to be manipulated, then add point cloud visualization in Rviz. Finally, we have an image viewer in a full screen GUI to output the image that the virtual camera sees to a projector. We implemented everything in ROS (Quigley et al. 2009). All of the files are available on GitHub¹, including a sample Rviz config file, sample point clouds in pcd format, and the launch files for publishing lens intrinsics and the pose of the projector.

Our projection mapping system functions on the principal that a projector is the dual of a camera. If a camera is thought of as a map from 3D world coordinates to 2D image plane coordinates, then a projector is a map from 2D image plane coordinates to 3D world coordinates. In a camera, each light ray from the world passes through the lens and hits the sensor. In a projector, each light ray passes through the lens and hits a surface in the world.

By using a virtual camera with the same intrinsics and extrinsics as the projector, we can then map points from the virtual world’s 3D space to 3D space in the real world.

3.1 Projector Selection Consideration

While any projector should theoretically work as long as we calibrate it to get the lens intrinsics as detailed below, there were a couple of factors that made us choose the ViewSonic PA503W projector in our implementation. In the common use case where projectors are used to watch movies and show presentation slides, the room lighting is often switched off or dimmed to make the projection more bright and thus more legible. However, robots often operate indoors with lights on and not dimmed, so the consideration here is to make the projection visible and legible even under bright light conditions.

There are three contributing factors: brightness, contrast and the projection technology. The standard measure of brightness is the ANSI lumens value, which is a measure of the total light output per unit of time. For reference, our PA503W produces 3,800 ANSI lumens of light. Contrast is expressed by a ratio between the darkest and lightest areas of an image, with our PA503W being capable of a 22,000:1 contrast ratio. Finally, there are two popular projection display technologies: Liquid Crystal Display (LCD) and DLP (Hornbeck 1997). As a brief summary from the work by Hornbeck, DLP is based on a Digital Micromirror Device (DMD), and it has a higher brightness than LCD because DMD has a reflective and high-fill factor digital light switch, as opposed to active-matrix LCDs, which are transmissive and thus the generated heat cannot be dispatched well. See (Hornbeck 1997) for more detail.

3.2 Projector Calibration

We modelled the projector with a pinhole lens model after determining the focal length f and principal point (c_x, c_y) . Both of these values were calculated manually by mounting the projector perpendicular to a flat surface at some fixed distance (e.g., 3), marking the corners of the image, and then



Figure 3: We calculated the intrinsics of our particular projector by mounting it perpendicular to a posterboard at a fixed distance. See Section 3.2 for more details.

using the projective equation:

$$f = w \frac{Z}{W} \quad (1)$$

where w is the width of the projected image in pixels, W is the width of the projected image in meters, and Z is the distance from the projector to the image plane in meters.

The principal point (c_x, c_y) can be calculated given X and Y , the respective distances from the origin to the intersection of the optical plane and the optical axis in meters:

$$c_x = w \frac{X}{W} \quad (2) \quad c_y = h \frac{Y}{H} \quad (3)$$

where h is the height of the projected image in pixels.

These values for f , c_x , and c_y were then placed into an intrinsic camera matrix K :

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

With this matrix calculated for our projector, we then publish it in a ROS CameraInfo message² and set up a virtual camera in RViz (see Section 3.3) that subscribes to the CameraInfo message and is located at a TF transform corresponding to the pose of the physical projector. The CameraInfo message for the ViewSonic projector and the details of this process are in our code³.

It is worth noting that measurement error may accumulate during this manual process. Since consumer projectors are typically designed with little radial or tangential distortion, we did not explicitly model lens distortion. In practice, the projection does not deviate much from the real objects, but it could be improved by more accurate projector-camera calibration methods (Moreno and Taubin 2012).

²https://docs.ros.org/melodic/api/sensor_msgs/html/msg/CameraInfo.html

³https://github.com/uml-robotics/projection_mapping/blob/master/projector_camera_info.yaml

¹https://github.com/uml-robotics/projection_mapping

3.3 Virtual Camera

We developed `rviz_camera_stream`⁴, an Rviz plugin that outputs an image of what a camera sees in the Rviz virtual world. We refer this as the *virtual camera*.

Represented in a ROS TF frame (Foote 2013), the pose of the virtual camera – placed in the virtual world in Rviz – is the same as the pose of the projector in the real-world. Together with the fact that the virtual camera pose is in the same transform hierarchy as a perception sensor, we are able to transform a point cloud or any other visualizations from the perception sensor’s frame to the virtual camera’s frame. This method allows the projection to be in a projector’s view point of what the perception sensor sees.

To have the same optical properties as the projector, the virtual camera subscribes to a ROS topic with the `CameraInfo` message mentioned in the previous subsection. This ensures that, when objects are projected back to the real-world, the projected objects match the object’s physical shape and size.

Finally, because this virtual camera resides inside Rviz, it is able to see everything being visualized in Rviz, such as interactive markers (Gossow et al. 2011), navigation maps, or point clouds. For example, point clouds are seen by the virtual camera and projected in Figure 1.

3.4 Projection Output

Because the `rviz_camera_stream` package publishes the image of what the virtual camera sees, we use the image viewer from the `image_view` ROS package to subscribe to the image topic and output to the projector through HDMI. Note that the image viewer is in full screen by enabling and using the “Toggle full screen” keyboard shortcut in the keyboard setting of Ubuntu’s Settings software.

3.5 Hardware Platform

As seen in Figure 4, we have demonstrated an implementation on a Fetch robot with a custom structure to mount a projector. While it would have been easier to attach the projector to the robot’s head, the neck may not have had enough torque to bear it, so we chose to attach the structure to the robot’s upper back. In order to pan and tilt the projector, we attached a ScorpionX MX-64 Robot Turret Kit⁵.

Despite using a Fetch robot, the projection mapping technique we describe is robot agnostic. Previously, we also applied it on an assistive robot (Wang et al. 2018). The only requirement is that there is a transform frame for the projector so it is integrated into the transform hierarchy of a robot. In our case, we created two frames: one for the projector lens and another for the attachment point at the bottom of the projector. However, this requirement does not necessarily mean that the projector must be attached to the robot, but instead that it must be co-located with the robot in the operating environment. For example, if one would like to use projection mapping with an industrial robot arm, the projector can be

⁴https://github.com/uml-robotics/rviz_camera_stream; thanks to Lucas Walter (<https://github.com/lucasw>) for his contribution.

⁵<https://www.trossenrobotics.com/p/ScorpionX-RX-64-robot-turret.aspx>



Figure 4: A sample use, where a projector is mounted onto a Fetch robot via a custom hardware structure attached to its upper back and a turret unit to pan and tilt the projector. However, the projection mapping technique is robot agnostic and the projector does not have to be attached to the robot. See Section 3.5 for more details.

placed nearby on a structure as long as its lens is pointing at what the perception sensor is targeting. By doing so, a robot arm will also not block the projection during manipulation, as careful readers may notice in the accompanying video. Otherwise, one can add a cylinder collision object to model the line of sight to mitigate blockage.

4 Conclusion

In this paper, we have detailed our approach to and implementation of projection mapping in robotics. Four major components were discussed: projector consideration and calibration, a virtual camera in Rviz, projection output, and a sample hardware platform.

To implement projection mapping, a roboticist can purchase an off-shelf projector, calculate a coordinate frame after installing it, calibrate its lens to get the lens intrinsics, and publish it in an encapsulated `CameraInfo` message. Then the roboticist would set up a virtual camera using the `rviz_camera_stream` Rviz plugin and subscribe to the `CameraInfo` message. Then the roboticist could add point cloud visualizations or any other displays (e.g., interactive markers) in Rviz and have the projector point them either manually or using a turret unit, and finally use an `image_viewer` to output the image from the virtual camera to the projector.

Acknowledgments

This work has been supported in part by the Office of Naval Research (N00014-18-1-2503). Thanks to Analog Devices (ADI) and MassRobotics for organizing the ADI Sensor Fusion Challenge⁶, providing funds to purchase the sensors. We also thank Brian Flynn for building the hardware structure and Daniel Giger for assembling the turret unit.

⁶<https://www.mradichallenge.com/>

References

- Admoni, H., and Scassellati, B. 2015. Robot nonverbal communication as an AI problem (and solution). In *2015 AAAI Fall Symposium Series*.
- Admoni, H., and Scassellati, B. 2017. Social eye gaze in human-robot interaction: a review. *Journal of Human-Robot Interaction* 6(1):25–63.
- Admoni, H.; Weng, T.; Hayes, B.; and Scassellati, B. 2016. Robot nonverbal behavior improves task performance in difficult collaborations. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, 51–58. IEEE.
- Allspaw, J.; Roche, J.; Lemiesz, N.; Yannuzzi, M.; and Yanco, H. A. 2018. Remotely teleoperating a humanoid robot to perform fine motor tasks with virtual reality. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*.
- Andersen, R. S.; Madsen, O.; Moeslund, T. B.; and Amor, H. B. 2016. Projecting robot intentions into human environments. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 294–301.
- Chadalavada, R. T.; Andreasson, H.; Krug, R.; and Lilienthal, A. J. 2015. 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)*, 1–6.
- Chakraborti, T.; Sreedharan, S.; Kulkarni, A.; and Kambhampati, S. 2018. Projection-aware task planning and execution for human-in-the-loop operation of robots in a mixed-reality workspace. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4476–4482. IEEE.
- Coovert, M. D.; Lee, T.; Shinde, I.; and Sun, Y. 2014. Spatial augmented reality as a method for a mobile robot to communicate intended movement. *Computers in Human Behavior* 34:241–248.
- Desai, M.; Kaniarasu, P.; Medvedev, M.; Steinfeld, A.; and Yanco, H. 2013. Impact of robot failures and feedback on real-time trust. In *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction*, 251–258. IEEE.
- Dragan, A. D.; Lee, K. C.; and Srinivasa, S. S. 2013. Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, 301–308. IEEE.
- Foote, T. 2013. tf: The transform library. In *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 1–6. IEEE.
- Gao, Y., and Huang, C.-M. 2019. PATI: a projection-based augmented table-top interface for robot programming. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 345–355.
- Ghiringhelli, F.; Guzzi, J.; Di Caro, G. A.; Caglioti, V.; Gambardella, L. M.; and Giusti, A. 2014. Interactive augmented reality for understanding and analyzing multi-robot systems. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1195–1201.
- Gossow, D.; Leeper, A.; Hershberger, D.; and Ciocarlie, M. 2011. Interactive markers: 3-D user interfaces for ROS applications. *IEEE Robotics & Automation Magazine* 18(4):14–15.
- Grundhöfer, A., and Iwai, D. 2018. Recent advances in projection mapping algorithms, hardware and applications. In *Computer Graphics Forum*, volume 37, 653–675. Wiley Online Library.
- Hornbeck, L. J. 1997. Digital light processing for high-brightness high-resolution applications. In *Projection Displays III*, volume 3013, 27–40. International Society for Optics and Photonics.
- Kwon, M.; Huang, S. H.; and Dragan, A. D. 2018. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 87–95. ACM.
- Moon, A.; Troniak, D. M.; Gleeson, B.; Pan, M. K.; Zheng, M.; Blumer, B. A.; MacLean, K.; and Croft, E. A. 2014. 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*, 334–341. ACM.
- Moreno, D., and Taubin, G. 2012. Simple, accurate, and robust projector-camera calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 464–471. IEEE.
- Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, 5. Kobe, Japan.
- Rosen, E.; Whitney, D.; Phillips, E.; Chien, G.; Tompkin, J.; Konidaris, G.; and Tellex, S. 2020. Communicating robot arm motion intent through mixed reality head-mounted displays. In *Robotics Research*. Springer. 301–316.
- Szafir, D.; Mutlu, B.; and Fong, T. 2015. Communicating directionality in flying robots. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 19–26.
- Wang, D.; Kohler, C.; Pas, A. t.; Wilkinson, A.; Liu, M.; Yanco, H.; and Platt, R. 2018. Towards assistive robotic pick and place in open world environments. *arXiv preprint arXiv:1809.09541*.
- Watanabe, A.; Ikeda, T.; Morales, Y.; Shinozawa, K.; Miyashita, T.; and Hagita, N. 2015. Communicating robotic navigational intentions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5763–5769.