

# Using 3D Mice to Control Robot Manipulators

Varad Dhat  
vdhat@uw.edu  
University of Washington  
Seattle, Washington, USA

Nick Walker  
nswalker@cs.washington.edu  
University of Washington  
Seattle, Washington, USA

Maya Cakmak  
mcakmak@cs.washington.edu  
University of Washington  
Seattle, Washington, USA

## ABSTRACT

Fluid 6DOF teleoperation of robot manipulators enables telemanipulation where autonomy is not possible, facilitates the collection of demonstration data, and aids routine robotics development. Amongst 6DOF input devices, 3D mice stand apart for their ergonomic design and low cost, but their sensitivity and users' relative inexperience with them require special design considerations. We contribute a web software package that makes integrating 3D mice in robot manipulation interfaces easy. The package consists of configurable input signal processing schemes that can make the device more forgiving by, for instance, rejecting small inputs or emphasizing a dominant axis, and an interactive visual representation of the device's 6DOF twist input, which helps with operator familiarization and provides a visual aide during teleoperation. We provide a demonstration interface illustrating a typical integration with a ROS/ROS2 robot system and give usage advice based on our research experience.

## CCS CONCEPTS

• **Human-centered computing** → **Haptic devices**; • **Computer systems organization** → **External interfaces for robotics**.

## KEYWORDS

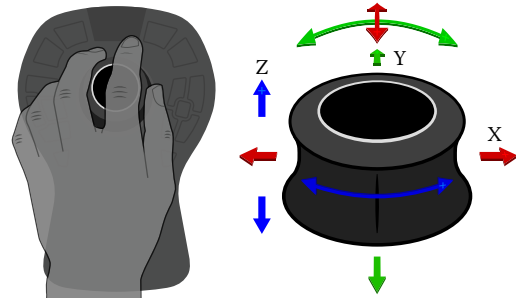
3D mouse, filtering, teleoperation, web

### ACM Reference Format:

Varad Dhat, Nick Walker, and Maya Cakmak. 2024. Using 3D Mice to Control Robot Manipulators. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24)*, March 11–14, 2024, Boulder, CO, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3610977.3637486>

## 1 INTRODUCTION

Robot teleoperation systems have become vital in sectors like healthcare, space exploration, and manufacturing [13, 17], where they enable colocated or remote operators to complete challenging manipulation tasks. While a rich literature has characterized the use of master-slave systems, where the operator interacts with a robot arm which supplies haptic feedback, there is now increasing interest in lower-end industrial or home use cases where less precision may be acceptable and a lower cost required. Cheaper teleoperation input devices such as teach pendants, joysticks and gamepads [21, 24] can be cumbersome and lack the degrees of



**Figure 1: The 3D mouse is a small desktop input device that supports simultaneous 3D translation and 3D rotation input.**

freedom to enable comparable fluid motions. One device that addresses these requirements is the 3D mouse [23]—often called a spatial mouse or a SpaceMouse, after the brand—which is explicitly crafted for navigating and interacting within three-dimensional spaces [10, 11]. Similar to a joystick, the device, shown in Figure 1, is attached to a fixed base and the user supplies a displacement and/or a rotation, usually with a single hand. This relative input requires only small movements from the operator, enhancing its ergonomics for extended use [7], compared to a master-slave system or a motion capture system [8].

We explore the 3D mouse for controlling the end-effector of a robot. Previous works have compared 3D mice to haptic controllers and have also been used for collecting data for imitation learning, indicating an increased demand for 3D mice in research applications [4, 5, 14, 25]. Numerous studies have demonstrated the utility of a 3D mouse in teleoperation [12, 15, 16, 22]. While these works have explored various tasks using the 3D mouse, there remains a gap in the availability of open-source software solutions. We address this gap by providing software designed to enhance 3D mouse-based teleoperation systems.

## 2 PURPOSE

The objective of our software package is to make 3D mice usable for robot operators. While the short throw of common 3D mice—a maximal translation displacement of 1mm and maximal rotation of 5deg—makes the devices suitable for extended use, it also makes them less forgiving for novice or less dexterous operators, and is the root of several common difficulties:

- (1) **Sensitivity:** Users are often surprised that the device begins registering input even for minute movements. Many find it difficult to rest their hand over the device without registering inadvertent input.

*HRI '24, March 11–14, 2024, Boulder, CO, USA*

© 2024 Copyright held by the owner/author(s).

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24)*, March 11–14, 2024, Boulder, CO, USA, <https://doi.org/10.1145/3610977.3637486>.

- (2) Dimensional coupling: The device is free to move in any dimension and has no bias towards particular axes, so users may struggle when attempting to isolate a dimension.
- (3) Limited feedback: The device’s mechanical design affords haptic feedback in the form of proportional counter-pressure, but the small scale of the feedback is often insufficient for new operators to interpret whether they are supplying the input they intended. Because this feedback is not controllable, 3D mice also cannot indicate to the operator that the robot controller is unable to accomplish the commanded input, due, for example, to joint limits, singularities or obstacles.

We address these issues with a configurable processing pipeline and a 3D visualization of the user’s input.

### 3 CORE FUNCTIONALITY

3D mice produce a 6-dimensional signal  $\mathbf{s} = [\mathbf{t}; \mathbf{r}]$  where vectors  $\mathbf{t} = [\ell_G \cdot \ell_- \cdot \ell_I]$  and  $\mathbf{r} = [A_G \cdot A_- \cdot A_I]$  represent translation and rotation displacement about axes of the device. Our package helps users understand their input and aids in the processing of the signal to produce a twist output  $\mathbf{S}$  consisting of angular velocities  $\mathbf{l}$  and linear velocities  $\mathbf{L}$  suitable for commanding a robot manipulator’s end effector.

*Input normalization.* We standardize the configuration of the input axes—X right, Y forward, Z up when seated behind the device—and scale values to be in the range  $[-1, 1]$  with the device rest state at 0, creating a uniform basis across varying device specifications.

*Button normalization.* Most 3D mice include one or more buttons, which can be used for arbitrary commands. Our package extracts, labels and forwards the state of the device’s buttons alongside the processed control signal.

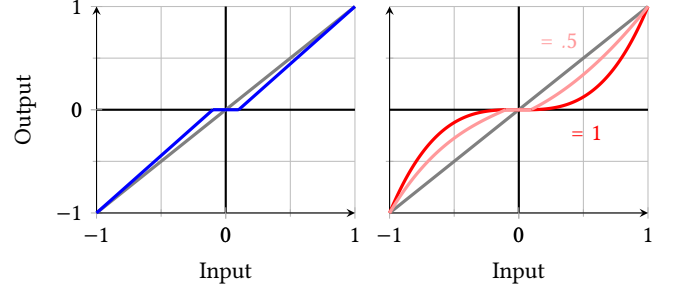
#### 3.1 Signal Processing

*Deadband.* Inputs within a zone  $[-3 \cdot 3]$ , with  $3 \in [0 \cdot 1]$  are nullified to reduce the occurrence of minor, unintentional inputs. This is accomplished with filter function

$$f(G) = \begin{cases} 0 & \text{if } |G| < 3 \text{ or } G = 0 \\ \frac{\delta(G) - \text{sign}(G)\delta(3)}{1 - \delta(3)} & \text{otherwise} \end{cases} \quad (1)$$

which removes values in the deadband, and applies a transformation  $\delta(G)$ , scaling the remaining values appropriately so that input begins smoothly at the boundary of the dead region. Figure 2 plots Equation 1 with an identity transformation and  $3 = 1$ .

*Input Sensitivity Curve.* Users often struggle to provide input in the lower range of the device when teleoperating small motions. In contrast, users rarely need to differentiate between similar near-maximal inputs. Following common practice in gaming applications, we redistribute input precision to favor smaller values using a cubic sensitivity transformation,  $\delta(G) = VG^3 + (1 - V)G$  [6]. The parameter of this operation,  $V \in [0 \cdot 1]$ , interpolates between cubic and linear functions. A value of 0 results in a scaled linear deadband, while values approaching one bow the line toward the x-axis, with a value of 1 resulting in a smooth (differentiable) transition across the deadband boundary. Figure 2 plots Equation 1 with cubic transformation and  $3 = 1$ .



**Figure 2:** Left: A linear scaled deadband rejects input in a small region around 0, and scales the remaining input range to cover the full output range. Right: A cubic scaled deadband acts similarly, but applies a cubic function to increase input sensitivity for smaller values.

*Dimensional Weighting.* Providing a bias in favor of axis-aligned unidimensional translation or rotation inputs helps novice operators overcome the difficulty of isolating dimensions. We provide a softmax reweighting filter which increases the contrast between small and large values in a vector  $\mathbf{x}$ , creating reweighted vector  $G_g^{\text{weighted}} = F_g G_g$  where

$$F_g = \frac{\exp(g^{-1}|G_g|)}{\sum_g \exp(g^{-1}|G_g|)} \quad (2)$$

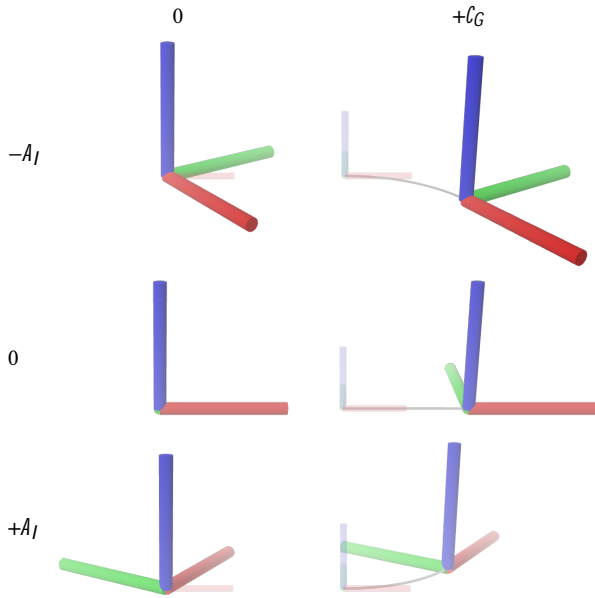
We apply this scheme to both  $\mathbf{r}$  and  $\mathbf{t}$ . If the device, for instance, registers a large value for translation component  $\ell_G$  and small values for  $\ell_-$  and  $\ell_I$ , the softmax function assigns greater weight to  $\ell_G$ , ensuring that the robot primarily moves in that direction. The degree of this emphasis is configurable using the temperature parameter  $g$ , resulting in no reweighting as  $g \rightarrow \infty$ , or in hard-max behavior as  $g \rightarrow 0^+$ . We preserve the magnitude of the input by scaling  $\mathbf{x}^{\text{weighted}}$  to have magnitude  $\|\mathbf{x}\|_1$ .

*Temporal Smoothing.* The near-instantaneous response of the device can make it feel jumpy to operators. Temporal filtering mitigates this issue, but by its nature, introduces a degree of latency to the control. We’ve found that an exponential moving average, also known as a low-pass filter, applied component-wise to  $\mathbf{s}$ , setting  $\beta_g^{\text{smooth}} = W\beta_g + (1 - W)\beta_g^{\text{prev}}$ , works well to address a range of situations with the appropriate configuration of decay weight  $W \in [0 \cdot 1]$ . Tasks which require smoother operation can set lower values, while those benefiting from maximal responsiveness can disable the filtering with a value of 1.

#### 3.2 Visualization

Limited haptic feedback is an obstacle to 3D mouse usability, so we designed a twist visualization to provide feedback tailored to several common user needs. Figure 3 shows the visualization in several configurations.

*Representing High Dimensional Input.* Users’ main challenge is understanding whether they are inputting what they mean to be inputting. The device provides counterpressure opposite in direction and magnitude to the user’s input, but it can be too subtle for some users. In order to represent both their 3D translation and



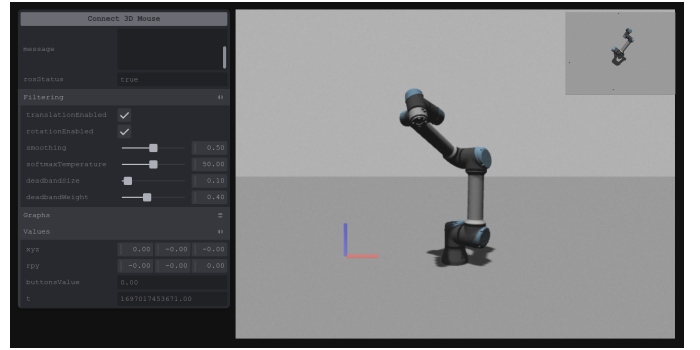
**Figure 3: The twist visualization used to provide feedback to operators, shown with different combinations of a translation (right column) and a rotation (bottom row).**

3D rotation input on a 2D screen, we render two axes markers in a 3D scene; a translucent anchor which serves as a reference point, and a moving target which represents their input, calculated as the pose  $target = \exp(S)$ . The target’s motion is scaled to be several times larger than the physical device’s true displacement to increase legibility. The scene is rendered using a perspective projection, causing parallel elements of the marker to converge to a point in the distance. To further enhance depth perception, the scene is rendered with “fog”; as the marker moves further from the camera, its color lightens and contrast is reduced, and as it moves closer, its colors darken and become richer.

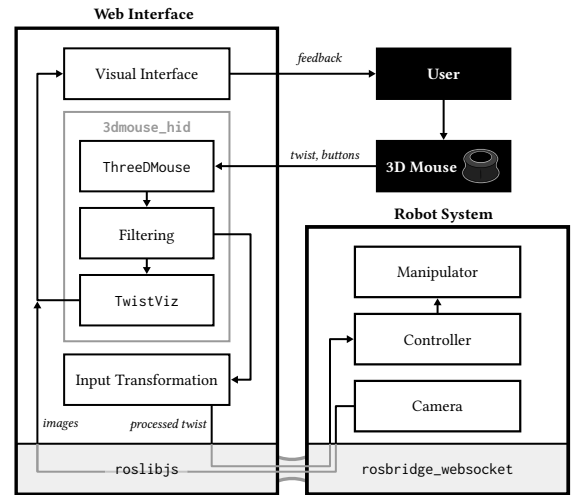
*Distinguishing Translation and Rotation.* Rotation control is often novel and challenging for users, who can struggle to understand whether and how much rotation they are applying. The orientation of the target marker with respect to the reference encodes this information, but to make it more direct, we render a “twist trace”; a cord which connects the origins of the axes markers. This line, calculated by stepwise integration of the twist, bends into an arc as rotation is increased, creating a clear visual signature for translation inputs which include rotation.

## 4 CODE

3dmouse-hid is listed on NPM, the major Javascript package index, and its repository is available at <https://github.com/hcrlab/3dmouse-hid> under the permissive BSD 2-Clause License. The core of the library consists of fewer than 1000 lines of code, and depends on only a single 3rd party package, factors which we believe will simplify community contribution and minimize maintenance burden.



**Figure 4: The included demonstration web-interface provides an example of how the package’s components can be integrated.**



**Figure 5: Software architecture in a typical usage**

## 5 USAGE NOTES

Installation instructions and inline documentation describe the details of our software, so in this section we highlight high-level usage considerations. The discussion in this section references an included teleoperation interface—shown in Figure 4—which serves as an illustration of a typical integration. The flow of information and division of responsibilities is charted in Figure 5.

*Browser Support.* Our package uses WebHID (Web Human Interface Device) to communicate with devices. Google Chrome and Microsoft Edge provide support, but WebHID remains a contested specification, and there are limited prospects for its adoption in other browsers [3].

*Device Support.* We include support for the current range of 3DConnexion SpaceMouse devices. Our code can work with any similar, HID-compliant device with the specification of button and axis bitfield mappings. The included specifications are provided in `deviceSpecs.js`. With further modification, this code could also

be used to interface with alternative input devices which use the HID specification, including commercially available sip-and-puff devices. Note that using lower dimensional input devices would require the design of a modal control interface or an inference-based assistive teleoperation system. While we do not include an example of such an interface in the package, there is significant prior work detailing the use of low dimensional input devices, such as 2D joysticks, to control 6D motion [1, 9, 19].

*Getting Values.* The `ThreeDMouse` class wraps the procedure for acquiring a `WebHID` device handle, receiving raw HID events, and processing them. It emits `3dmouseinput` events which carry a timestamped frame of filtered output.

*Buttons.* Applications will have varying uses and needs for buttons, so we do not attempt to prescribe any functionality or filtering for them. In our applications, we have found it useful to map buttons to UI functions like changing the input mode or swapping camera views, or to simple robot functionality like homing the robot or toggling the gripper. More advanced uses would benefit from application-specific button state management and debouncing.

*Filtering.* `ThreeDMouse` can be configured with a filter object which can apply arbitrary operations to device signals before they are emitted. We provide a `SmoothingDeadbandFilter` class which exposes the full set of parameters described in Section 3.1, making it configurable for many applications. The appropriate filtering configuration depends on the task and the users. While it is likely not appropriate to expose the full bank of parameters directly to users, it may be valuable to offer a range of pre-tuned profiles or a calibration procedure to ensure broad usability.

*Mapping Input to Robot Motion.* We’ve found that transforming the device’s twist ( $\tau$ ) to a robot frame for control can require careful design consideration to match operator expectations. In our demonstration remote teleoperation interface, we provide a front facing view which tilts slightly downward to show the plane the robot rests on. Given such a view, users generally expect the axes of control to align with the axes of the robot’s base. They implicitly map the X and Y axes in the base frame to the X and Y axes of the device, aligning them so that the motion of the device corresponds to the motion of the robot in the given camera view. Although “up” from the camera’s perspective is tilted slightly away from the gravity axis, users still expect control to occur aligned to gravity. The interaction between the scene context, the position of the robot with respect to the camera, and varying user and task needs means that the input transformation needs to be designed per application.

*Robot Controller.* 6DOF input is most suitable for end-effector velocity controllers, which directly accept 6DOF twist goals. In our teleoperation setup, we have used `MoveIt Servo` [2], which converts twist commands to joint velocities using differential inverse kinematics.

*Orienting the Visualization to Match Robot Motion.* In order for the `TwistViz` visualization to serve as an effective feedback mechanism, it must always be perceived as matching the physical movement of the device and the movement of the robot, however this becomes impracticable when multiple, disparate camera views are used (as is commonly necessary to handle occlusion). Movement of

the device can only naturally correspond with movement of the robot in a single view, so the input transformation must be modal and change according to the active view. It’s possible for operators to become disoriented if they are controlling the robot while looking at a view that does not match the active input transformation. In our example interface, we create a clear visual hierarchy amongst views and we draw the twist visualization on top of the primary view to emphasize their connection.

## 5.1 Ethical Considerations

This package and other tools for remote teleoperation risk promoting the thoughtless removal of colocated operators, which can negatively impact safety by increasing worst case latency of human intervention. Further, applications of remote teleoperation may function to reduce worker power by dispersing human workforces and shifting them to areas with weaker labor protections.

## 5.2 Responsible Use Guidelines

Our package may exacerbate safety issues with teleoperation systems if misconfigured or applied without appropriate safeguards. In particular, failure to configure appropriate output scaling may lead to the generation of large velocity commands. We recommend adopting a compliant controller, which can guarantee the robot will not exert excessive forces regardless of operator input. Basic safety limits like velocity and acceleration limits must also be enforced to protect the robot’s actuators.

It is up to users to ensure that their interfaces adhere to web accessibility best practices [18] and to provide sufficient alternative input schemes for users with limited dexterity [20].

## 5.3 Utility and Relevance

The 3D mouse is the most widely available and most cost effective 6DOF input device. Our package democratizes its usage in robot teleoperation, addressing common issues facing operators with the use of signal processing and visual feedback. For researchers, this package may be useful in systems for Programming by Demonstration (PbD) or Learning from Demonstration (LfD), where it can facilitate data collection by reducing familiarization time for novice operators. The device’s ergonomic benefits compound with usage duration, so it may be suitable for use by professional operators supervising or teaching fleets of industrial manipulators. It may also be useful for bimanual teleoperation, where the cost and ergonomic benefits are doubled.

Our package also fills a void for researchers already using 3D mice, providing a well-known reference implementation against which they can compare their implementations and—we hope—to which they can contribute their ideas and improvements. While more complex than day-to-day input devices, by providing a common core implementation we can promote effective, reproducible use of 3D mice in robotics research.

## ACKNOWLEDGMENTS

This work was supported in part by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) Grant #1R01EB034580-01, National Science Foundation (NSF) grant IIS-1924435, and the UW/Amazon Science Hub.

## REFERENCES

- [1] Maria E. Cabrera, Tapomayukh Bhattacharjee, Kavi Dey, and Maya Cakmak. 2021. An Exploration of Accessible Remote Tele-operation for Assistive Mobile Manipulators in the Home. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. 1202–1209. <https://doi.org/10.1109/RO-MAN50785.2021.9515511>
- [2] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. 2014. Reducing the barrier to entry of complex robotic software: a moveit! case study. *Journal of Software Engineering for Robotics* (2014). [https://doi.org/10.6092/JOSER\\_2014\\_05\\_01\\_p3](https://doi.org/10.6092/JOSER_2014_05_01_p3)
- [3] Mozilla Contributors. [n. d.]. Mozilla's Positions on Emerging Web Standards. <https://github.com/mozilla/standards-positions>
- [4] Carolin Fellmann, Daryoush Kashi, and Jessica Burgner-Kahrs. 2015. Evaluation of input devices for teleoperation of concentric tube continuum robots for surgical tasks. In *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, Robert J. Webster III and Ziv R. Yaniv (Eds.), Vol. 9415. International Society for Optics and Photonics, SPIE, 94151O. <https://doi.org/10.1117/12.2076741>
- [5] Anton Franzluebbers and Kyle Johnson. 2019. Remote Robotic Arm Teleoperation through Virtual Reality. In *Symposium on Spatial User Interaction* (New Orleans, LA, USA) (*SUI '19*). Association for Computing Machinery, New York, NY, USA, Article 27, 2 pages. <https://doi.org/10.1145/3357251.3359444>
- [6] Mimir Games. 2017. Joystick input and using deadbands. <http://www.mimirgames.com/joystick-input-and-using-deadbands-article/>
- [7] Soheil Gholami, Marta Lorenzini, Elena De Momi, and Arash Ajoudani. 2022. Quantitative Physical Ergonomics Assessment of Teleoperation Interfaces. *IEEE Transactions on Human-Machine Systems* 52, 2 (2022), 169–180. <https://doi.org/10.1109/THMS.2022.3149167>
- [8] Soheil Gholami, Francesco Tassi, Elena De Momi, and Arash Ajoudani. 2021. A Reconfigurable Interface for Ergonomic and Dynamic Tele-Locomotion. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4260–4267. <https://doi.org/10.1109/IROS51168.2021.9636775>
- [9] Laura V Herlant, Rachel M Holladay, and Siddhartha S. Srinivasa. 2016. Assistive teleoperation of robot arms via automatic time-optimal mode switching. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 35–42. <https://doi.org/10.1109/HRI.2016.7451731>
- [10] G Hirzinger, B Brunner, K Landzettel, N Sporer, J Butterfass, and M Schedl. 2003. Space Robotics—DLR's Telerobotic Concepts, Lightweight Arms and Articulated Hands. *Autonomous robots* 14 (03 2003), 127–45. <https://doi.org/10.1023/A:1022275518082>
- [11] Gerd Hirzinger, Max Fischer, Bernhard Brunner, Ralf Koeppel, Martin Otter, Markus Grebenstein, and Ingo Schäfer. 1999. Advances in Robotics: The DLR Experience. *The International Journal of Robotics Research* Special Issue on Robotics at the Millenium (11 1999), 1064–1087. <https://doi.org/10.1177/02783649922067726>
- [12] Yoonho Kim, Emily Genevriere, Pablo Harker, Jaehun Choe, Marcin Balicki, Robert W. Regenhardt, Justin E. Vranic, Adam A. Dmytriw, Aman B. Patel, and Xuanhe Zhao. 2022. Telerobotic neurovascular interventions with magnetic manipulation. *Science Robotics* 7, 65 (2022), eabg9907. <https://doi.org/10.1126/scirobotics.abg9907> arXiv:<https://www.science.org/doi/pdf/10.1126/scirobotics.abg9907>
- [13] Jing Luo, Wei He, and Chenguang Yang. 2020. Combined perception, control, and learning for teleoperation: key technologies, applications, and challenges. *Cognitive Computation and Systems* 2, 2 (2020), 33–43. <https://doi.org/10.1049/ccs.2020.0005>
- [14] Ajay Mandekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. 2018. ROBOTURK: A Crowdsourcing Platform for Robotic Skill Learning through Imitation. In *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings (Proceedings of Machine Learning Research, Vol. 87)*. PMLR, 879–893. <http://proceedings.mlr.press/v87/mandekar18a.html>
- [15] Marcus Mast, Zdeněk Materna, Michal Španěl, Florian Weisshardt, Georg Arbeiter, Michael Burmester, Pavel Smrž, and Birgit Graf. 2015. Semi-autonomous domestic service robots: Evaluation of a user interface for remote manipulation and navigation with focus on effects of stereoscopic display. *International Journal of Social Robotics* 7 (2015), 183–202. <https://doi.org/10.1007/s12369-014-0266-7>
- [16] Zdeněk Materna, Michal Španěl, Marcus Mast, Vítězslav Beran, Florian Weisshardt, Michael Burmester, and Pavel Smrž. 2017. Teleoperating Assistive Robots: A Novel User Interface Relying on Semi-Autonomy and 3D Environment Mapping. *Journal of Robotics and Mechatronics* 29, 2 (2017), 381–394. <https://doi.org/10.20965/jrm.2017.p0381>
- [17] Luka Peternel and Arash Ajoudani. 2023. After a Decade of Teleimpedance: A Survey. *IEEE Transactions on Human-Machine Systems* 53, 2 (2023), 401–416. <https://doi.org/10.1109/THMS.2022.3231703>
- [18] The A11Y Project. 2024. Checklist. <https://www.a11yproject.com/checklist/>
- [19] Camilo Perez Quintero, Oscar Ramirez, and Martin Jägersand. 2015. VIBI: Assistive vision-based interface for robot manipulation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 4458–4463. <https://doi.org/10.1109/ICRA.2015.7139816>
- [20] Vinitha Ranganeni, Noah Ponto, and Maya Cakmak. 2023. Evaluating Customization of Remote Tele-operation Interfaces for Assistive Robots. In *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 1633–1640. <https://doi.org/10.1109/RO-MAN57019.2023.10309395>
- [21] Beibe Shu, Halldor Arnarson, Björn Solvang, Tero Kaarlela, and Sakari Pieskä. 2022. Platform independent interface for programming of industrial robots. In *2022 IEEE/SICE International Symposium on System Integration (SII)*. 797–802. <https://doi.org/10.1109/SII52469.2022.9708905>
- [22] Jörn Vogel, Katharina Hertkorn, Rohit U. Menon, and Máximo A. Roa. 2016. Flexible, semi-autonomous grasping for assistive robotics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 4872–4879. <https://doi.org/10.1109/ICRA.2016.7487692>
- [23] Weidong Xiang, Ling Long, Jiale Yuan, and Liyuan Liu. 2021. A 3D Mouse Design based on A Three-axis Acceleration Gyroscope. In *2021 International Conference on Networking, Communications and Information Technology (NetCIT)*. 167–171. <https://doi.org/10.1109/NetCIT54147.2021.00041>
- [24] Fengxin Zhang, Chow Yin Lai, Milan Simic, and Songlin Ding. 2020. Augmented reality in robot programming. *Procedia Computer Science* 176 (2020), 1221–1230. <https://doi.org/10.1016/j.procs.2020.09.119> Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.
- [25] Yuke Zhu, Ziyu Wang, Josh Merel, Andrei Rusu, Tom Erez, Serkan Cabi, Saran Tunyasuvunakool, János Krámar, Raia Hadsell, Nando de Freitas, and Nicolas Heess. 2018. Reinforcement and Imitation Learning for Diverse Visuomotor Skills. In *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania. <https://doi.org/10.15607/RSS.2018.XIV.009>