Eye Gaze for Assistive Manipulation

Reuben M. Aronson

February 18, 2020

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Henny Admoni, *chair*
Artur Dubrawski
Nancy Pollard
Brenna Argall, *Northwestern University*

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

Copyright © 2020 Reuben M. Aronson. All rights reserved.

For humans.

Abstract

Full robot autonomy is the traditional goal of robotics research. To work in a human-inhabited world, however, robots will often need to collaborate with humans. For example, many scenarios require human users to teleoperate robots to perform tasks, a paradigm that appears everywhere from space exploration, to disaster recovery, to assistive robotics. This collaboration enables tasks to be performed more smoothly or safely than humans could without requiring full robot autonomy. However, robots are hard to control. To compensate, roboticists build shared control systems, in which the robot operator's command is combined with an autonomous plan to accomplish the operator's goal.

We propose to enhance shared control systems by observing people's natural, nonverbal behavior and using that signal to gain additional insight into their goals and concerns about the task. In particular, how people look at a scene depends on what they are thinking about the scene. Research into eye gaze behavior shows that when manipulating objects by hand, people look at their next goal or next obstacle as they become relevant and rarely at task-irrelevant places. By tracking and processing a user's eye gaze behavior, shared control systems can build more complex models of the user's intentions. Equipped with this knowledge, shared control systems can both provide more accurate assistance and new types of assistance.

In this thesis, we begin by conducting a study examining how people's eye gaze behavior relates to the task performance while they teleoperate a robot manipulator. Next, we develop a pipeline for processing the raw eye gaze sensor signal to include task context and develop models to learn aspects of user mental state from this gaze signal. Finally, we design and evaluate two gaze-based assistance systems: goal recognition, which we compare with input-based goal recognition strategies; and dynamic concern-based collision avoidance, a new approach in shared control. This thesis establishes the usefulness of the eye gaze signal for enabling more sophisticated shared control behaviors. Moreover, it shows how monitoring people's natural behaviors can be incorporated into human-robot collaboration for more sophisticated mental state modeling and corresponding behavior.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Monitoring Natural Eye Gaze Improves Teleoperation

The promise of fully autonomous robots capable of performing complex tasks based only on high-level instructions persists as a valuable goal for robotics research. However, robots are capable of providing significant benefits even without full automation: through teleoperation, robots can perform tasks controlled by a human operator. This control strategy sidesteps the distant goal of full autonomy and instead develops a system that works, in which robots can perform simple tasks and a user provides the complex sensing and planning systems required. Indeed, teleoperated systems are used widely in deployed robots today, for such tasks as space exploration, disaster recovery, or assisted manipulation. Moreover, even if we can achieve full autonomy, teleoperation nevertheless remains a valuable control design strategy applicable to many scenarios. For example, high-risk activities (such as surgery or space operation) may be too dangerous to leave to fully autonomous systems, or personal activities (such as bathing) may be more comfortably performed without the perception of a separate autonomous agent.

However, teleoperated systems are difficult to control. Specifically, accomplishing manipulation tasks through teleoperation typically requires simultaneously controlling several degrees of freedom while adapting to (possibly dynamic) contact with the environment. Depending on the domain, this problem is made further complicated by

issues such as controller latency, nonintuitive controller-to-end-effector mappings, and the limitations of visual feedback due to occlusion and the inability to perceive forces. The range of proposed solutions is similarly various, including novel interface design (such as whole-arm monitoring or force-feedback joysticks) and controller design (to ensure stability under latency). For applications such as assistive robotics, however, these interfaces cannot be used due the physical capabilities of the intended users. Therefore, alternative approaches are required.

One approach that has developed recently to ease teleoperation is *shared control*. In these systems, a robot arm is controlled simultaneously by a user (via joystick or other input device) and an autonomous planning system. The planner recognizes the operator's goal and plans a trajectory for the robot to take. Then, the controller and planned trajectories are combined to accomplish the task. This approach does not rely on sophisticated physical interfaces and is therefore especially promising for assistive manipulators. Indeed, it has shown some promise in that domain by considering only the user's controller inputs for goal recognition [4, 5].

We can do even better. People continuously emit natural, nonverbal signals telegraphing their intentions, concerns, and mental state in general. Without increasing the control difficulty, shared control systems can derive a much richer understanding of their controller's view of the task by monitoring these signals. This rich model of the user's mental state while performing a task enables more sophisticated shared control behavior.

To achieve this more sophisticated user model, we turn to a new sensing paradigm: eye gaze detection. Psychological research in eye gaze behavior indicates that eye gaze is strongly connected to task progression during manipulation [6, 7, 8, 9]. People look at the objects they are reaching for and at the obstacles they are avoiding. People also look at tasks differently depending on their expertise in accomplishing the task [10], their facility with the signal input system [11], their cognitive load [12], etc. In addition, people produce this signal naturally and unconsciously. Unlike gaze-based input systems, using eye gaze in this way does not disturb the user's ability to interact with the task as they normally would, and it requires no additional training. By detecting people's eye gaze behavior while they are teleoperating a robotic manipulator, we can build a much more complex model of their approach to the task and provide correspondingly complex assistance behavior.

1.2 Application to Assistive Systems

To make the problem specific, we focus on a particular application for teleoperated manipulators: wheelchair-mounted assistive robots for people with upper mobility disabilities. For decades, assistive robot manipulators have been an area of exciting research [13, 14], and using these robots allows people to accomplish a wide variety of tasks [15, 16, 17]. Making these arms more usable directly contributes to improving the quality of life for people with certain disabilities.

Assistive systems are particularly appropriate for shared control approaches. This domain shares many of the same challenges of other teleoperation tasks in requiring complex motion and contact with the environment. The usual control input in this domain, in which users control two dimensions of end-effector motion at a time using a joystick, only compounds these challenges [18]. However, approaches that work well for other task domains do not necessarily transfer well for assistive robotics. Since people use assistive robot arms to accomplish a wide variety of activities of daily living, relying on specific assumptions about desired robot behavior or task domain does not extend well. In addition, novel control interfaces such as whole-arm sensors are often unusable by people with disabilities. Therefore, a shared control approach is particularly appealing for this domain. Indeed, this strategy has already shown success. One implementation [4] enabled non-disabled users to complete their tasks faster and more efficiently, and another study with users with disabilities [5] showed that users do prefer to work with some amount of this type of assistance.

Improving on shared control by incorporating eye gaze inputs is especially important for the assistive domain. Since people who use assistive robot arms use them frequently to perform many different tasks, even minor increases in efficiency compound to become larger quality-of-life increases. Providing natural eye gaze requires no special actions by the user, since people already produce these signals automatically. In addition, eye gaze grants access to the user's perception of the task, which can enable complex forms of assistance without requiring a fully general solution for robotic manipulation.

1.3 Contributions

In this thesis, we demonstrate how by monitoring the natural eye gaze behavior of people performing a manipulation task with a robot, we can achieve more sophisticated techniques for shared control. To begin, we conducted a study of eye gaze behavior during teleoperated manipulation and compared these behaviors with those shown during by-hand manipulation. We found that people indeed look at their manipulation targets and obstacles that are causing them problems, though they look much more at the robot end-effector then at their hands. People's glances at their goal objects are correlated with which part of the task they are performing, and their glances towards unlikely objects like internal robot joints correspond to times when those joints cause task failure. In addition, we collected a dataset of eye gaze and user behavior during this teleoperation task. Finally, we propose to repeat the eye gaze study with people with disabilities to understand how their gaze behavior aligns with that of non-disable users. These observations and this dataset enable the development of assistive algorithms.

Next, we develop a pipeline to process the raw eye gaze signal to incorporate contextual information about the task. Eye gaze trackers typically provide a pixel location on an egocentric video scene as their output. While many eye gaze analysis strategies work on this signal directly, activity recognition during manipulation is heavily dependent on which objects the user looks at. Therefore, mapping this raw gaze signal to semantic object labels can provide better results [19, 20]. However, this semantic labeling problem is difficult, since the task is fully three-dimensional and both the user's head and the task objects move throughout the task. We present a strategy for incorporating the contextual information into gaze and an approach that improves the labeling accuracy on a real dataset. This process shows a general way to transform a gaze signal for any manipulation behavior into a form appropriate for different types of activity recognition.

Finally, we present two new approaches for assistance based on this extended knowledge of the user's mental state. For the first approach, we will explore using eye gaze behavior to derive the user's goal. Specifically, we will develop machine learning models that predict the user's goal from the semantic eye gaze labels emitted by our gaze processing pipeline. We will compare this method with methods that derive goal

predictions from the user control input alone, and we will evaluate a fully working system in a user study. For the second approach, we introduce a new assistance strategy, in which we use eye gaze behavior to control how the robot navigates around an obstacle. People look more often towards an obstacle that they are concerned about, so a robot can adapt its safety boundary based on how much the user looks at that obstacle. We will develop a model for determining object concern from eye gaze behavior and build this assistance type into our shared control implementation. Finally, we will evaluate the usefulness of this novel assistance behavior in a user study.

Chapter 2

Background

2.1 Shared Control Paradigms

To ease the problem of robot control, many approaches have been presented to fuse the user's input with an autonomously generated signal. One category of assistance consists of stateless assistance: a robot behavior that can be determined directly from the robot position, environment, and unchanging elements of the task. In this type of assistance, no updating human model is used. Perhaps the most straightforward example of this approach is given by Ramacciotti et al. [21], which proposes an assistive welding system in which the tool frame motion is divided between the user and the robot control. For example, the robot maintains forward motion in x while the user controls the other translational axes y and z. The motion provided by the assistance is determined beforehand by the task parameters and does not change as the task progresses. Similarly, Vu et al. [22] reorients how the user controls map into the rotation axes of the tool frame. This reorientation is a static alteration of the control scheme that does not vary by task circumstance. A more complex type of motion assistance is virtual fixtures [23], which modify the compliance of the robot controller based on its position and the intended direction of motion. For example, motion along the $\pm x$ axis proceeds easily, whereas motion in other directions either moves more slowly (in the open-loop case, where the control gain is reduced) or results in a restoring force back to the desired motion surface (in the closed-loop case).

Another set of stateless control schemes aid the operator in avoiding obstacles.

Crandall and Goodrich [24] presents a system that uses a variant of potential fields to automatically maintain distance from known obstacles, and it shows that adding this automatic obstacle avoidance behavior enables users to focus on another task while controlling a robot. You and Hauser [25] compares several strategies for fusing obstacle avoidance with user input commands. They test three categories of obstacle avoidance: end-effector control with collision rejection, in which commands that lead to detected collision are ignored; potential field control, in which the user command is modeled as an attractive force and environment obstacles as repulsive forces to maintain distance from obstacles; and full motion planning control, in which after receiving a motion command, the robot autonomously plans a collision-free path to the goal position using an RRT and executes that motion. These avoidance techniques all lead to faster task completion with fewer collisions. While these strategies all vary substantially in purpose and complexity, they can all be implemented without using any variable models of human state.

More sophisticated assistance behaviors can be achieved by explicitly modeling otherwise-invisible parts of the operator's internal state. In stateful assistance, systems maintain models of aspects of the user's intentions and update them over time. Aarno and Kragic [26] presents a system for recognizing low-level motions ("gestemes") using layered hidden Markov models. If an operator is trying to move the device in a circle, for example, the HMM can recognize this gesture automatically and provide assistance to maintain it. In a more extensible example, Hauser [27] uses a dynamic Bayes net to infer a user's task over a wide variety of task definitions. The assistance system maintains a distribution over likely tasks and enacts different assistance categories based on that task recognition.

Often, the internal state required by such assistance is the user's goal, represented as a final robot position. Systems predict the user's own goal among a finite set of choices by autonomously generating plans to achieve each goal and then comparing the user's actual control input with the generated plans. This approach works when the observations used to infer the goal are exactly the user input commands, and it has the advantage that the computed task solution can often be reused in generating an assistive command. This approach has been used to control wheelchairs based on planner results [28] or based on modelling the deviations of the user's actual provided command from a nominal user model [29, 30, 31]. While these models provide motion

based on the best-matched user goal according to the user model, more complex uses of the user goal matching are possible. Dragan and Srinivasa [32] proposes thresholding the user goal matching probability and only providing assistance when the system confidence in its model of the user goal exceeds a given value. Building on this approach, shared autonomy [4, 33] plans assistive actions over the user model uncertainty, which allows the algorithm to provide useful assistance even when exact goals are not known.

Once an overall shared control system has been selected, the implementation details are still important to refine. Gopinath et al. [5] ran a study in which users manually controlled the amount of assistance provided by a shared control system. Users did use the assistive control, indicating that these methods are useful. However, they also kept the assistance lower than optimal task completion time would achieve, indicating that users balance their desire for control with using assistance to reduce the task time. Dragan and Srinivasa [32] has success with modulating the amount of assistance based on the algorithm's confidence in its model of the user's goal. Gopinath and Argall [34] change the starting joystick mode so that the user's initial input is maximally informative about the user's goal.

While assistance behaviors that do not model mental state achieve success, adding explicit user models enables more sophisticated behaviors. Our work builds on these assistance frameworks by using the more complex mental state representations that eye gaze enables to build more sophisticated assistance behaviors.

2.2 Eye Gaze

In this thesis, we present the idea of monitoring people's eye gaze behavior in order to recognize their intentions during teleoperated manipulation. To understand how eye gaze behavior relates to people's intentions during manipulation, and how to process that signal, we turn to existing eye gaze research.

2.2.1 Eye Gaze During Manipulation

Eye gaze behavior during by-hand manipulation has been a subject of study in psychology for over twenty years. Johansson et al. [6] describes a study in which

users were instructed to grasp an object, manipulate it around an obstacle, and place it down elsewhere. While performing that task, people followed consistent eye gaze patterns. They looked at relevant locations before interacting with them: people look at the object until just before they grasp it, the obstacle until just before navigating around it, and the placement location just before placing the object. In addition, the paper found that people rarely looked at their own hands and that their gaze was almost entirely directed towards task-relevant locations. Similar results were found with people performing natural tasks like making tea [8] and making a sandwich [9]. Perhaps most similar to a teleoperation task is the experiment reported in Sailer et al. [11], in which users are given a non-intuitive mouse controller and instructed to move a cursor around a screen. While learning the controls, people watched the cursor motion on the screen and briefly glanced at the goal positions; once they were comfortable, people's eye gaze behavior looked more similar to that in by-hand manipulation.

2.2.2 Eye Gaze for Intent Recognition

Since eye gaze behavior is so closely tied to people's goals, it has been widely studied as a modality for understanding people's mental state in a variety of applications. We restrict this review to analysis techniques more closely related to our manipulation task; see Lukander et al. [35] for a full review. Bader et al. [36] had people perform manipulation actions on a screen simulating a table and used gaze patterns to predict people's next behavior (reaching, moving, or releasing) and intended object. Matsuzaka et al. [37] showed that people's gaze predicts their intended grasp object and strategy (one- or two-handed) in a VR manipulation task. In a human-robot interaction study, Huang and Mutlu [38] uses hand-crafted features to predict a person's food order from a manipulator robot. Duarte et al. [39] shows that people can follow gaze cues when seeing other people perform an object manipulation task, and their understanding persists even when a robot is giving gaze cues.

There are several eye gaze analysis strategies available for performing different kinds of intention recognition. Analysis of the eye gaze dynamics directly without scene context has been successful at tasks such as identifying whether people are performing free viewing or visual search [40] or which of several different tasks a

person is performing [41, 42, 43, 44]. However, this type of context-free analysis performs poorly when trying to recover specific information about *how* a task is performed [19, 20]. For tasks such as predicting gaze behavior during walking [45], driving [46, 47], or combined walking and object manipulation in VR [48], general gaze-based saliency features about the scene were much less effective than modeling the dynamics of the actual task.

For predicting specific information about people's behavior during a task, one approach often used is *scanpath analysis* [49]. In this method, the eye gaze signal is treated as timeseries data rather than being reduced to frequency data. Kübler et al. [50] quantized scanpaths into a small set of regions and used lexical analysis to predict if people will pass a driving test. Kubler et al. [51] goes beyond this approach by dynamically clustering fixations based on SIFT features around the point-of-regard and uses these sequences to determine if people are performing a tea-making task for the first or second time. Chen and Ballard [52] uses a hidden Markov model trained on timeseries gaze and hand position signals to predict which stage of a letter stapling task the participant is executing. We build on these scanpath techniques by modifying them for a manipulation task to use the additional semantic context available.

2.3 Teleoperated Robots With Eye Gaze

Now that we have described how the eye gaze signal reveals mental state in general, we describe how it has been used for some robotic systems.

2.3.1 Eye Gaze as Direct Input

There has been some research in using eye gaze for direct robot control. The usual strategy presented is to use the user's eye gaze as a primary input device for an autonomous manipulator system [53, 54, 55, 56, 57, 58]. In these systems, people look at an object they wish to grasp, and the robotic system performs object recognition, maps the gaze to an object in the scene, and autonomously grasps it. Similar systems have been presented for wheelchair navigation [59] or mobile robot navigation [60]. In addition, Tong et al. [61] present a scheme for using gaze location as a set point

for a controller during remote surgery, and McMullen et al. [62] use gaze as an input method to a screen controller which is paired with a brain-computer interface directing a robot arm. While eye gaze can be used as a direct input as seen here, our approach instead monitors the user's *natural* eye gaze behavior while completing a task.

2.3.2 Natural Eye Gaze for Shared Control

Rather than using eye gaze as direct input, in this thesis we propose using natural eye gaze as an indirect input for shared control. This idea has been discussed by Admoni and Srinivasa [63], which proposes to infer the user's goal based on how close their gaze is to each of the possible goals, and Nikolaidis et al. [64], which proposes a framework for modeling user intention from gaze with naive Bayes updates. Possibly the most similar work published recently is Stolzenwald and Mayol-Cuevas [65], in which people operate a handheld controller to interact with objects on a screen. The user's natural eye gaze behavior is used to predict which object they will interact with next, and they show that assisting towards or against that goal influences the user's task success. We build on these approaches by using more knowledge about the eye gaze signal and the dynamics of the scene to more accurately infer the user's goal. Moreover, we demonstrate additional categories of assistance that rely on additional inference from gaze beyond just looking at the user's goal.

Chapter 3

Eye Gaze and Mental State

3.1 Introduction

To build assistive systems based on eye gaze behavior, we must first understand how people use their eyes during teleoperated manipulation. By understanding how people's eye gaze corresponds to their mental state and the mechanics of the task, we can build systems that use the gaze signal to determine when and how to apply assistive controls. Understanding this connection is foundational to developing novel assistance behaviors.

We start by conducting a user study to capture eye gaze during teleoperated manipulation. We asked 24 users to perform a food acquisition task by teleoperating a robot arm while we recorded their eye gaze patterns. We found that while people look primarily at the end-effector of the robot, they also look at their goal objects and places that cause problems, which matches with results from by-hand manipulation. Next, we repeat the study with higher quality eye gaze sensors to build a data set of gaze behavior during teleoperation, which we have made available publicly. Finally, we propose to conduct a similar study with people with upper mobility impairments who would be users of this arm in collaboration with the Human Engineering Research Laboratories at the University of Pittsburgh. This last study will validate that assistance based on our research will remain useful for likely users of assistive robot arms.

3.2 Completed Work: Eye Gaze During Teleoperated Manipulation

3.2.1 Introduction

For this study, participants were asked to perform an eating task with a robot arm while their eye gaze behavior was recorded. We found that while people often look at the end-effector of the robot, they still look at their goal objects periodically during *planning* and *monitoring* glances, which follow particular patterns of the task. In addition, we describe instances when users encounter task problems and their eye gaze behavior changes according to the problems they face. These two observations will serve as the basis for our assistive strategies.

3.2.2 Study Methodology

For eye gaze and assistance validation, we use a robot eating task. In this task, participants teleoperate a robot to pick up one of three morsels on a plate in front of them (see Fig. 3.1). This task, introduced by Javdani et al. [4], satisfies several goals: it allows for clear, symmetric goal selection (among the offered morsels), and it is a simplification of an eating task, which users with disabilities have indicated is among the most important functions for such an arm to perform [66]. We use this task for the studies here as well as later on (with some adaptation) to validate our proposed assistance strategies.

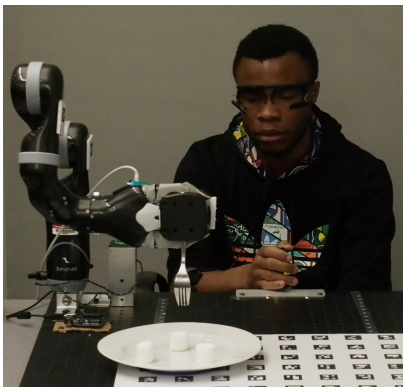


Figure 3.1: The eating task.

During the task, participants sit facing a table. In front of them are a robot arm (a Kinova Mico [67]) mounted to the table and a plate holding three morsels of food (we use marshmallows for their ease of spearing). Participants began the study (after informed consent) by receiving an explanation of how to control the robot and then spending 5 minutes practicing with it. Then, for each trial, participants were instructed to select and share which morsel they intended to target. They then used the robot to move the fork held by the robot to a position above their target morsel. They then pressed a button on the joystick to complete the task, at which time the robot autonomously moved down to the morsel, then it moved the fork towards the user's mouth to simulate most of an eating motion. The actual spearing was done autonomously so that the minimum robot height could be restricted and we could avoid table collisions.

During each trial, the user controls the robot using a two-axis joystick via modal control, which is typical for these types of arms. During modal control, the two-axis joystick maps to successive pairs of degrees of freedom of the end-effector (x/y, z/yaw, pitch/roll; see Fig. 3.2). Pressing a button on the joystick cycles through the modes. The robotic system can process this joystick input directly into end-effector commands or add an assistance strategy. The five minute practice period helps participants to understand this control strategy, though it remains difficult for many users.

(a) x-y mode (b) z-yaw mode (c) pitch-roll mode

Figure 3.2: Modal control for joystick-controlled manipulators. The user controls two axes of motion at a time with a joystick and uses a button to cycle through the three modes. Figures from Aronson et al. [1].

Participants performed this task five times each for four different assistance conditions, fully counterbalanced over 24 participants. In this study, the assistance

conditions consisted of full teleoperation, in which the users had complete control; shared autonomy, a policy-based assistance blending strategy (elaborated on in Sec. 5.2); policy blending, an alternate, more conservative assistance strategy; and full autonomy, in which the robot selected a morsel and planned a trajectory itself while ignoring all user input. While eye gaze was collected for all conditions, we focused on the teleoperation and shared autonomy assistance conditions primarily, as they represent the conditions most related to our intended use. Eye gaze data was collected over a total of 120 trials per condition. Eye gaze data was collected using a Pupil Labs Pupil monocular eye tracker [68]; see Sec. 4.2.1 for additional information about how this sensor is used.

3.2.3 The HARMONIC Data Set

After publishing the results of the previous study, we decided to repeat the study in order to amass a higher quality data set that would be appropriate for quantitative analysis. To do so, we repeated the study described above, but with a few enhancements. First, we upgraded the eye gaze sensor to a binocular sensor, which gives much higher quality gaze sensing for three-dimensional gaze. We also recorded more of the internal gaze sensor data for later postprocessing. Second, we altered the assistance algorithm slightly, by replacing the *shared autonomy* and *blend* conditions with two different levels of shared autonomy, and replacing the *autonomous* condition with a mode in which goal intention was derived from the user input but actual control direction was supplied entirely by the autonomous system. Finally, we added an electromyography sensor on the user's wrist. This data set has been made publicly available at `harp.ri.cmu.edu/harmonic` and is under review; a preprint is available on arXiv [69].

Once the raw data was collected, we segmented all of the gaze data from the teleoperation runs into distinct fixations and manually matched to each fixation which of the objects in the scene the participant was looking at (scene objects are shown in Fig. 3.3). Four coders labeled the fixations with one of ten labels or *unknown*. Twelve randomly-selected trials (10% of the data) were coded by all coders, and the average pairwise Cohen's kappa (inter-reliability rating) was 64.5% indicating good agreement. We expand more on the details of this labeling process in Chap. 4.

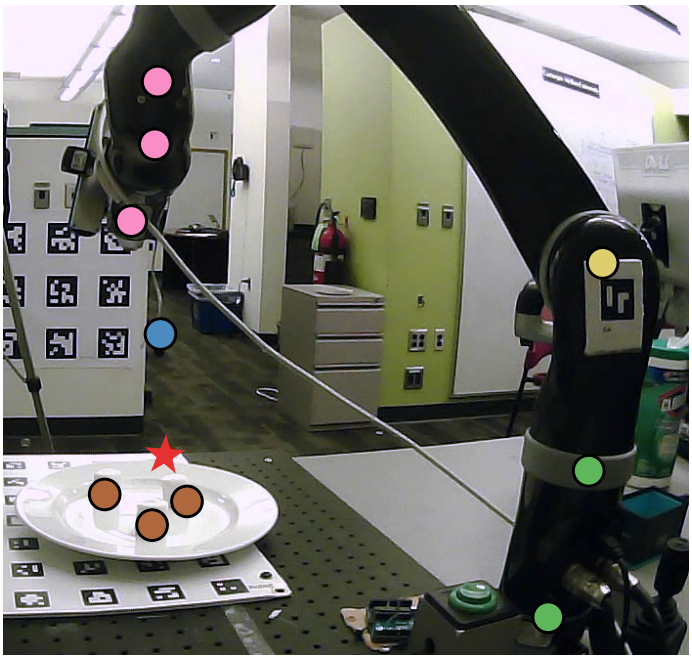


Figure 3.3: Fixations were manually assigned to one of ten scene keypoints, which were the three goal morsels and each robot joint. For bulk comparison, scene keypoints were also grouped as shown in the colors of the scene. Figure from Aronson and Admoni [2].

3.2.4 Eye Gaze Behavior Results

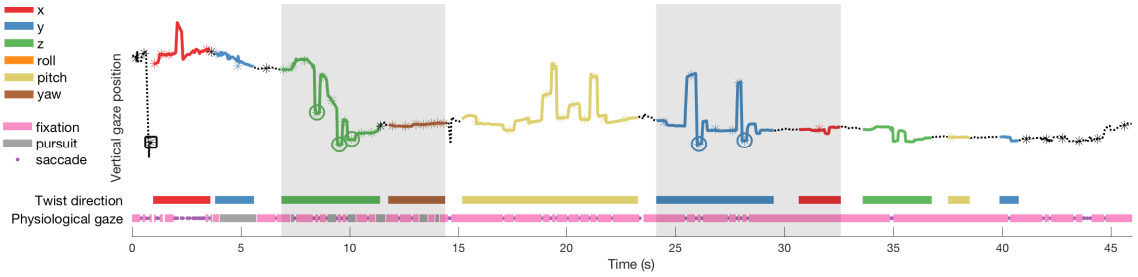


Figure 3.4: Vertical position of gaze points in the world image over time from a representative trial. Twist direction colors indicate which DOF is being controlled by the participant through the joystick; physiological gaze colors and dots indicate detected fixations, smooth pursuits, and saccades (see Sec. 4.2.2). Plate glances are outlined with either a black square (planning glance) or colored circle (monitoring glance). Shaded sections highlight two examples of repeated monitoring glances. (Figure and caption from Aronson et al. [1]).

To understand the broad patterns of eye gaze behavior during teleoperation, we examine the participant's eye gaze behavior during one teleoperated trial (Fig. 3.4). In this run, the participant begins by performing a *planning glance*: the user looked at the end-effector of the robot, then their target object, then back to the end-effector, all without moving the robot. Next, the participant moves the robot in the x/y mode to the approximate location of the end-effector above the morsel, and they watch the end-effector through the entire process. The participant then toggles to the z/yaw mode and lowers the robot in z, while performing a *monitoring glance*: alternating focus between the end-effector and the target while the robot is moving. Then, the participant aligns the fork vertically above the morsel while moving in yaw, pitch, and roll; they look at different places on the end-effector but do not glance at their target. Finally, the participant performs fine alignment in x, y, and z, performing *monitoring glances* throughout.

From this example, we derive some generalizations about eye gaze during teleoperated manipulation:

People spend a lot of time looking at the end-effector of the robot. Unlike in by-hand manipulation, people look at the end-effector of the robot throughout the trial. Specifically, $68.1 \pm 2.1\%$ of the fixations during each trial were at the end-effector or tool. Presumably, this gaze difference is due to people needing visual feedback to determine the location of the robot end-effector, whereas during by-hand manipulation, people can use their own proprioception to determine their hand position.

People look at their goals based on the status of the task. As indicated above, two eye gaze patterns recurred: *planning glances*, in which people held the robot stationary and alternated their focus between the end-effector of the robot and their goal object, and *monitoring glances*, in which people moved the robot while looking back and forth between it and their goal position. These patterns were frequent, with planning glances appearing in 76% of trials. In addition, morsel monitoring glances were significantly more frequent during translation than during rotation (Fig. 3.5). Repeated morsel monitoring glances, in which participants checked the morsel position more than once while they watched the robot end-effector, also

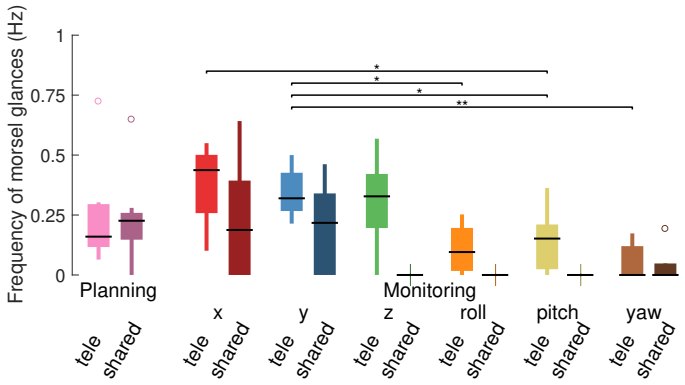


Figure 3.5: Mean frequency of planning and monitoring glances to the plate during each robot assistance mode. Monitoring glances are subdivided by joystick control direction. * indicates significance at the $\alpha = 0.05$ level; ** at $\alpha = 0.01$. Figure and caption from Aronson et al. [1].

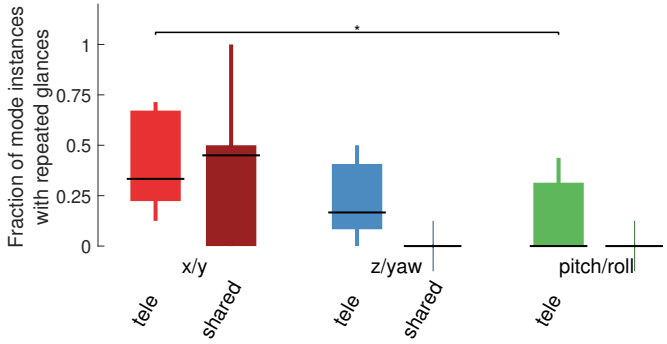


Figure 3.6: Proportion of joystick control sequences of the same mode that contained multiple (≥ 2) monitoring glances, subdivided by their control mode. * indicates significance at the $\alpha = 0.05$ level. Figure and caption from Aronson et al. [1].

occurred more often when in the x/y translation mode than in the pitch/roll pure rotation mode (Fig. 3.6). This distinction between rotation and translation may be because participants find rotation harder [70] or because an external reference for the target is less necessary during rotation. In both cases, we find that the timing of meaningful plate glances is highly related to the dynamics of the task.

Another key insight from this study is how people behave when they encounter problems in the task [3]. While these incidents occur infrequently in the dataset, we can nevertheless examine some case studies to understand how people's eye gaze changes when something goes wrong. To illustrate this phenomenon, we describe two

case studies that appeared in our HARMONIC data set (see Sec. 3.2.3).

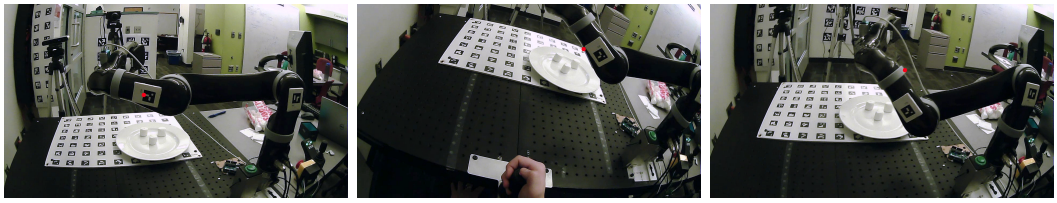


Figure 3.7: When the robot occludes the goal morsels, people move their heads for a better view. Figures modified from Aronson and Admoni [3].

People move their heads to compensate for robot occlusion (Fig. 3.7). In several cases, people moved the robot into a configuration where the robot itself occluded their view of the target morsel. In these situations, people often moved their heads significantly more than usual in order to get a better view. While this pattern is not revealed through semantic gaze analysis (see Sec. 4.3), it can be obtained from the raw gaze signal and the head motion. Knowledge that the operator is struggling to see a particular object can be used to supply useful contextual assistive actions.

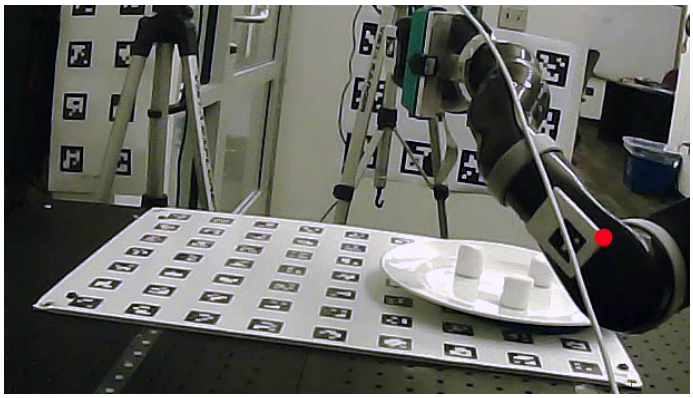


Figure 3.8: When the robot moves into a problematic joint configuration, people look at the joint that is causing problems. The red dot represents the participant's gaze location. Figure modified from Aronson and Admoni [3].

People look at robot joints during kinematic failure (Fig. 3.8). As noted above, people often look at the end-effector of the robot. People rarely look anywhere else on the robot, except for one notable case: when the robot goes into a problematic

kinematic configuration, people will look at the joint that is causing them an issue. For example, the robot has two general configurations while being teleoperated downward, which we can call *elbow-down* and *elbow-up* (see Fig. 3.8). In the *elbow-down* configuration, the robot cannot be moved towards the table, as the robot joint collides with the table before the target position is reached. It is difficult to fix this problem with only end-effector control, as the motion required is by definition within the nullspace of the joint Jacobian. When this problem occurs, people often look at the location of the joint that is causing them a problem. Again, with contextual information, this eye gaze pattern can indicate to an assistive system that corrective behavior would be especially useful.

3.2.5 Conclusion

In this section, we discuss a general framework for understanding human eye gaze during manipulation, either by robot or by hand: *people look at objects that are important to the task*. This pattern develops differently between manual and teleoperated manipulation, but shares some themes:

- People look at the end-effector of the robot most of the time (though they do not look at their own hands).
- People look at the target of their manipulation at times related to the progress of the task.
- People look at the locations of potential failure, whether obstacles (in manual manipulation) or kinematic failures (in teleoperated manipulation).

These results build a foundation for using eye gaze as a signal for assistive manipulation. By validating that people indeed look at their goal objects while teleoperating a robot, we show that this signal can be used to predict the operator's goal. Therefore, we can compare the use of eye gaze for goal prediction with the input-based strategies used by other assistance approaches. In addition, we find that people look at the scene differently in different failure situations, so eye gaze can enable additional assistance strategies beyond just goal assistance. Finally, the data set collected and the qualitative patterns found provide a platform for algorithmic processing of the eye gaze signal.

3.3 Proposed Work: Eye Gaze in Patients with Upper Mobility Impairments

While the study described above revealed compelling patterns in the eye gaze behavior of robot arm operators, it was only conducted on non-disabled users. Since the ultimate target of this assistance system is to be used by people with disabilities that prevent them from having full mobility, it is necessary to ensure that these patterns recur in that population. To verify that the eye gaze patterns that we described are representative, we propose to run another data collection study with participants drawn from a population of likely arm users.

To perform this study, we are building a collaboration with the Human Engineering Research Laboratories (HERL) at the University of Pittsburgh. HERL has extensive experience working with people with mobility disabilities, including assessing the robot we have used in various tasks [71]. We have been working with Dr. Joshua Chung, a postdoc in the lab, to put together an IRB and prepare a study.

We anticipate that the study will answer the following research questions:

RQ1 What patterns of eye gaze behavior occur during teleoperated manipulation by SCI users? Which of the patterns described in manual and teleoperation by non-disabled users recur, and which differ?

RQ2 What are the systematic challenges with collecting and analyzing eye gaze behaviors in this community? Are there aspect of the disability that makes the signal particularly noisy or different from that of non-disabled users?

RQ3 How do these users perceive assistive behaviors such as shared autonomy?

To investigate these questions, we will repeat the study detailed above with modifications appropriate to the new population. While the modifications will be developed in collaboration with HERL, we anticipate having to resolve several questions. We must determine clinical standards to measure level of disability and determine appropriate participants. We also need a plan for adapting our study for use with people who come with their own wheelchairs. We anticipate resolving these questions in collaboration with HERL and through piloting with the population.

3.4 Conclusion

In this chapter, we discussed our study of eye gaze behavior during teleoperated manipulation. These insights lay the groundwork for using gaze behavior for shared control.

3.4.1 Completed Work

The original study and eye gaze results were presented at HRI 2018 [1]. The examples of anomalous eye gaze behavior correlating to task failure were collected at the Fundamentals of Joint Action workshop at RSS 2018 [3]. The HARMONIC data set [69] is under review.

3.4.2 Proposed Work

One additional study will be conducted to determine the applicability of the above results to the target population. This study will be conducted in collaboration with HERL at the University of Pittsburgh. The study itself is well developed, as it is similar to the one already conducted, though modifications will need to be made for the change in population and location. The collaboration is currently working on preparing an IRB. We anticipate that this work will result in one paper published at a robotics and assistance conference such as ASSETS or HRI.

Chapter 4

Eye Gaze Analysis Pipeline

4.1 Introduction

To use our insights about eye gaze behavior to enable assistive algorithms, we must be able to automatically collect and analyze the eye gaze signal. Eye trackers, by default, emit a 2D (or 3D) coordinate that represents their user's point of regard relative to a scene camera. While using this raw signal directly enables some gaze-based inference [40, 41, 42, 43, 44], determining specific information about task parameters is difficult without contextual information about the task. Therefore, we process this signal to incorporate this task information.

To process the signal, we make some assumptions:

Users focus on one object at a time. In particular, people look directly at objects of interest. This assumption generally holds during by-hand manipulation [6, 72, 9]. While peripheral vision does provide some assistance for by-hand manipulation [73], the central assumption that gaze is task-directed is strongly supported in the psychology literature [19].

Users look at known objects. Manipulation tasks tend to involve interacting with a limited number of objects that stay unchanged during the task. Since glanced objects are task-relevant [8, 9], we assume that the the majority of informative glances are to the set of objects relevant to the manipulation task. Any off-object glances are

assumed to be noise or a signal that (for example) the user is not paying attention to the task, so they can safely be combined into a single category.

These two assumptions enable us to incorporate context into the gaze signal by labeling gaze data with the object that the user is most likely looking at during that time. Then, rather than raw gaze data, we use sequences of object labels as input to intent inference systems. Since assistive manipulation systems usually require their own object detection, object tracking is already present in the system. This processing pipeline converts the raw gaze data to a format where context is already included, which eases the development of intent inference systems.

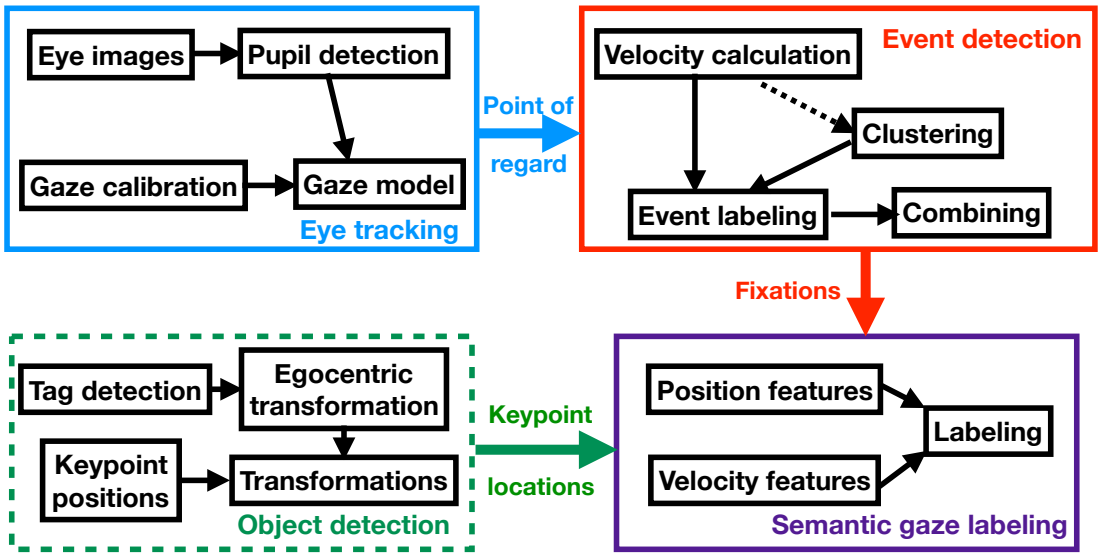


Figure 4.1: Flow chart of the gaze analysis pipeline. Raw eye gaze is subdivided into fixations and then labeled with a keypoint supplied by an (external) object tracking system.

The processing pipeline proceeds through several steps (Fig. 4.1). First, the raw eye gaze data is collected using an off-the-shelf eye tracker. Next, the eye gaze data

is segmented into fixations, corresponding to physiological principles of eye gaze and easing the classification problem. Then, each individual fixation is labeled with the most likely object in the workspace that it corresponds to. Finally, the timed sequences of foveated objects is analyzed according to the needs of our assistance procedure.

4.2 Background: Collecting the Eye Gaze Signal

4.2.1 Eye Gaze Sensors

Developments in eye tracking technology over the last several years mean that tracking gaze direction in a real-world environment at high sample rates is now possible. While significant research has been done using remote eye trackers that determine where people are looking on a screen, for our application is it most appropriate to use a *mobile eye tracker*, such as the Pupil Core [68] or the Tobii Pro Glasses 2 [74] (Fig. 4.2). These systems consist of a glasses-like frame worn by the user, on which a number of cameras are mounted. One or two IR cameras are mounted above the users' eyes (corresponding to a monocular or binocular setup) and record high-frequency video of the eyes themselves. In addition, a forward-mounted ("egocentric" or "world") camera captures the scene from the point of view of the user.

(a) Pupil Core binocular eye tracker. (b) Tobii Pro Glasses 2 binocular eye tracker.

Figure 4.2: Mobile eye trackers.

The first step in the sensor pipeline is to identify the center point of the user's

pupils on the eye camera image. Typical algorithms for pupil detection include simple region and curvature detection algorithms or more sophisticated 3D eyeball models. These models also emit pupil detection confidence values (between 0 and 1), which can be used to filter out poor-quality data or user blinks. In any case, this algorithms is usually packaged with the eye tracker itself. In all the work discussed here, we used the Pupil Labs Pupil tracker with their built-in pupil detection.

Next, the pupil center points must be mapped to the gaze point in the world camera. This mapping is typically found by performing a calibration procedure, in which the user is instructed to look at a camera target in the workspace and an operator moves the target around. For screen calibration, it is typical to perform a nine-point calibration procedure, in which the tag is moved to each point in a 3-by-3 grid encompassing the region of interest. For 3D eye tracking, we use a 27-point calibration procedure, where the tag is moved to each point in a 3-by-3-by-3 grid encompassing the workspace of interest. Recording calibration data at different depth planes is necessary for obtaining good results in 3D [75]. While improving the calibration fitting process is an active area of research [76], we use the off-the-shelf method provided by the eye tracker. Once the calibration data is collected, a model (typically a 2D polynomial) is fitted to the eye and world camera data. Care must be taken to ensure that the user's head moves as little as possible during calibration: the calibration is most accurate for interpolation among the calibration points, so their range must cover the user's workspace. Motion of the head during calibration can lead to reduced coverage in the gaze calibration space. Once again, these calibration algorithms are typically bundled with the eye tracker itself. In this work, we use the 27-point calibration available as "manual marker calibration" on the Pupil Labs software.

The output of the mobile eye tracker includes two 120Hz streams of pupil pixel location in the eye cameras and one 30Hz stream of gaze target pixel location in the world camera. This raw data is processed in the rest of our pipeline.

4.2.2 Event Detection

To simplify the process of identifying what objects people are looking at, we can take advantage of some of the physiological characteristics of human gaze. Rather

than having free control over eye gaze direction, people follow consistent eye gaze patterns, which consist primarily of *fixations* ($\approx 500 - 2000$ms stationary periods) separated by *saccades* (rapid $\approx 100 - 500$ ms ballistic trajectories between fixation locations). Controlled eye gaze motion typically only occurs in two situations: *smooth pursuits*, when someone looks at a moving object and follows it with their eyes, and *vestibulo-ocular reflex* (VOR), when people move their heads and their eye gaze moves to compensate while focusing on the same object. Therefore, when determining what object people are looking at, we need not perform the classification problem at the sample rate of the eye gaze sensor. Instead, we can perform saccade and fixation segmentation (known in the literature as *event detection*) and assume that during a fixation, smooth pursuit, or vestibulo-ocular reflex, the entire period is spent focusing on the same object.

There are two traditional algorithms for event detection: dispersion thresholding (I-DT) and velocity thresholding (I-VT) [77]. Both depend on noting that the point-to-point velocity during saccades are generally much larger than during fixations (or VOR, or pursuits). In I-DT, a measure of dispersion (e.g. variance) is calculated over windows of the eye gaze signal. Windows less than a manually-chosen value are determined to be fixations, while windows above the value are labeled saccades. In I-VT, the point-to-point velocity of the signal (the numerical derivative) is computed, and each point is labeled a fixation or saccade if it is below or above a custom threshold. Then, successive fixation labels that exceed a minimum fixation time are fused together and determined to be a single fixation.

For this work, we use a variant of I-VT, known as I-BMM [78, 79]. This method similarly calculates the velocity of the eye gaze signal (by angle), but learns a dynamic threshold by fitting a 2-component Gaussian mixture model to a sample of eye gaze data. Then, adjacent fixation labels are fused, and fusions that exceed a specified minimum time are labeled as fixations. A python implementation that works both offline and online was written and made open-source[1].

Our event detection algorithm varies slightly from the standard eye tracking approach due to the dynamics of our task. First, we do not distinguish between fixations, pursuits, and vestibulo-ocular reflexes. In the eye gaze signal, fixations appear as periods of zero velocity, whereas pursuits and VORs appear as periods

[1] http://github.com/HARPLab/ibmmpy

of small but nonzero motion. However, since all three categories involve the user focusing on a single object, for labeling objects of interest the distinction is not important. Therefore, in our fixations, there may be some internal motion, which may make distinguishing between saccades and fixations more difficult. However, saccade motions are still significantly significantly faster than pursuit or VOR motions, and I-BMM still gives good results. Even if pursuits are split into several subsequent fixations, the labeling procedure (below) should still label them the same, and a corrective procedure at the end can fuse adjacent fixations on the same object into a single label.

4.3 Completed Work: Semantic Gaze Labeling

One of the challenges of extracting information from eye gaze behavior is incorporating the scene context into the signal. While using the raw dynamics of the signal has been effective for such problems as activity recognition (visual search vs. scene viewing) [80, 36] or expertise measurement [10], the information we are concerned with is intrinsically tied with the specific objects of the task. Therefore, we choose to first process the data to incorporate semantic information about what objects people are looking at.

To incorporate this semantic information, we assign to each fixation a label indicating which of a pre-selected set of task objects the user is looking at. This approach makes the gaze data motion- and calibration-independent, which should enable more accurate processing. We also improve on the several existing algorithms that use 3D object tracking for fixation identification [81, 82, 83, 84, 85, 86] by presenting *velocity features* (Sec. 4.3.3 and Aronson and Admoni [2]), which track the motion of the objects to improve labeling accuracy.

4.3.1 Formal definition

Formally, we describe the problem as such:

Given: a sequence of eye gaze locations segmented into fixations $I_t = (i^t_0, \cdots, i^t_{m_t})$, where I_t is an index set representing a consecutive subsequence in τ of the original gaze sample g_τ. These fixation subsequences are derived using an event detection

τ	Index into individual eye gaze samples (typically 30Hz)
g_τ	Raw gaze sample, as a pixel in an egocentric camera
t	Index into individual fixations
$I_t = (\tau_0^t, \cdots, \tau_{m_t}^t)$	Index set representing fixation t
k_τ^i	Keypoint location for keypoint i at sample time τ
k_t^i	Mean value of the keypoint location during fixation t, $\text{mean}_{\tau \in I_t} k_\tau^i$
ℓ_t	Assignment of fixation t to one of the keypoints $i \in (1 \cdots n)$
c_t	Mean value of the gaze location during a fixation, $\text{mean}_{\tau \in I_t} g_\tau$
p_t^i	Value of position feature corresponding to keypoint i for fixation t
δf_t	Fixation-to-fixation difference between fixation $t-1$ and fixation t
δk_t^{ij}	Difference between the location of keypoint i during fixation $t-1$ and the location of keypoint j during fixation t
v_t^i	Value of velocity feature corresponding to keypoint i for fixation t

Table 4.1: List of symbols used for semantic gaze labeling.

algorithm, as described in Sec. 4.2.2.

Given: A set of timeseries keypoint locations k_τ^i, determined from an object detection algorithm. Each keypoint k^i is a semantically relevant object in the workspace (as determined manually by the experimenter).

Goal: Assign to each fixation t a label $\ell_t \in (1, \cdots, n)$ representing which keypoint the user is likely to be looking at for that particular fixation. Then, the gaze can be represented as a sequence $(f_1, \cdots, f_n)$ where $f_t = (\ell_t, d_t)$ represents both the fixation's label and its duration ($d_t = \tau_{m_t}^t - \tau_0^t$).

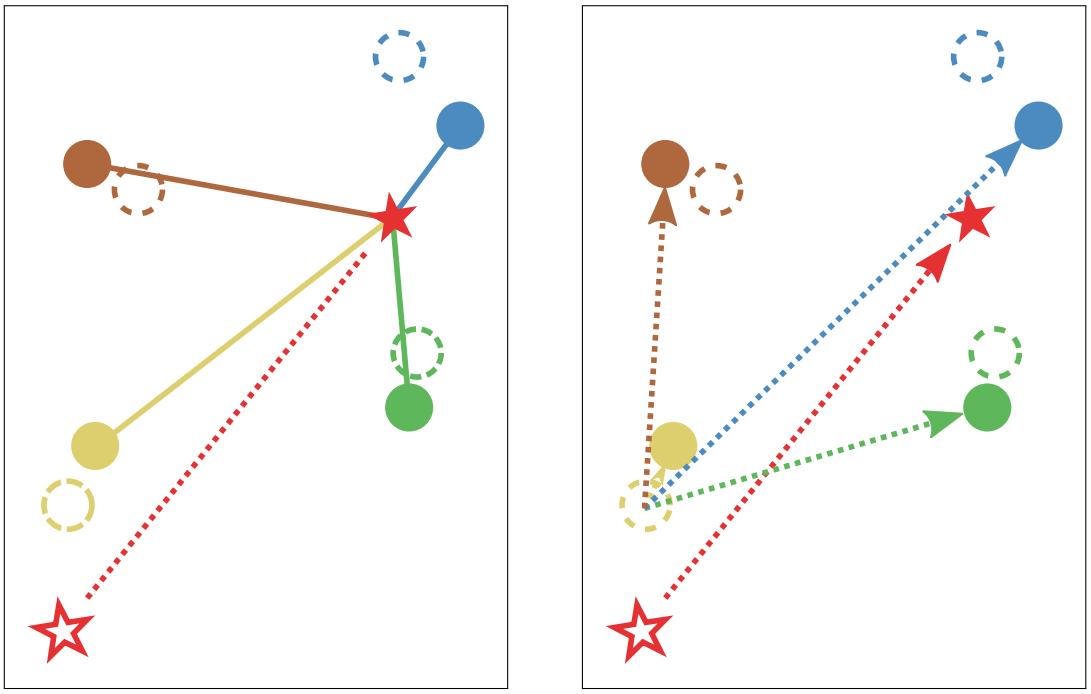
4.3.2 Position features

One straightforward way to compare the fixation subsequence to each keypoint to determine how well they align is to use the distance between them averaged over the entire fixation (see Fig. 4.3a). In particular, let

$$c_t = \text{mean}_{\tau \in I_t} g_\tau$$

represent the average gaze point during the fixation, and

$$k_t^i = \text{mean}_{\tau \in I_t} k_\tau^i$$



(a) Position features. (b) Velocity features.

Figure 4.3: A schematic representation of the calculated features. Colored circles represent keypoints. Filled circles represent keypoint positions at the current time. Outlined circles represent keypoint positions at the previous time, with the solid outlined circle representing the keypoint that was assigned as label; the other previous keypoints are dashed. The filled red star represents the average fixation location at the current time, and the outlined star the previous fixation location. Figure 4.3a represents the position features at the current time; the closest keypoint to the fixation is the blue one, but the distance is similar to the green distance due to a constant offset. Figure 4.3b represents the velocity features; the relative motion that the fixation would have taken between the previous time and the current time is represented by a dashed arrow for each keypoint and the observed relative motion by the dashed red arrow. The high similarity between the blue arrow and the red arrow leads to a small velocity feature for the blue keypoint independent of the constant offset. (Figure and caption from Aronson and Admoni [2].)

represent the average keypoint location for each keypoint i during each fixation t.

In the case of an actual fixation, the gaze is roughly stationary and this sequence should have small variance. During pursuit or VOR, the point will move during the sequence so the average is a poor measure, but by assumption the corresponding

keypoint moves similarly, so any error induced by taking the mean will be matched by the error in the keypoint.

Once the means are calculated, we can determine a position feature by computing the distance between the fixation mean and the position mean,

$$p_t^i = d_p(c_t, k_t^i), \tag{4.1}$$

where d_p is a distance function over gaze points g_τ. (See Appendix A for details on calculating this distance function).

4.3.3 Velocity features

One issue with the position features described above is that they tend to be highly susceptible to position error. Many of the errors in the gaze signal, such as calibration error, appear as large slow-changing position errors. To counteract the effect of constant errors, we can draw inspiration from signal processing and take the derivative of the comparison (see Fig. 4.3b). In particular, if we assume that

$$c_t = k_t^{\ell_t} + \epsilon_t,$$

where ϵ is a roughly constant error term ($\frac{\partial \epsilon}{\partial t}$ is small), then if we subtract the same equation for the previous fixation we get

$$c_t - c_{t-1} = k_t^{\ell_t} - k_{t-1}^{\ell_{t-1}} + (\epsilon_t - \epsilon_{t-1}).$$

By assumption, $\epsilon_t - \epsilon_{t-1} \approx \frac{\partial \epsilon}{\partial t}\Delta t$ is small. Therefore, if ℓ_t is the correct label,

$$(c_t - c_{t-1}) - (k_t^{\ell_t} - k_{t-1}^{\ell_{t-1}}) \approx 0,$$

so this feature value should be smaller for correct values of ℓ_t. Thus, we want to compare how the change of gaze target *between fixations* compares to the change of keypoint locations between fixations.

Formally, define

$$\delta f_t = \overrightarrow{d_p}(f_{t-1}, f_t)$$

to represent the vector change between f_{t-1} and f_t. (See Appendix A for a definition of $\overrightarrow{d_p}$). Then, we can determine the vector change between keypoints i and j during fixations $t-1$ and t respectively by computing

$$\delta k_t^{ij} = \overrightarrow{d_p}(k_{t-1}^i, k_t^j).$$

Finally, we compute the velocity feature v_t^i that measures how well fixation t matches keypoint i as

$$v_t^i = d_v(\delta p_t, \delta k_t^{\ell_{t-1} i}),$$

where ℓ_{t-1} represents the label (keypoint index) assigned to the previous fixation and d_v is a distance function over velocities defined in Sec. A. Note that this velocity feature term relies on the label assigned to the previous fixation, ℓ_{t-1}. This dependency makes this feature vulnerable to stacking error: if the previous label ℓ_{t-1} is incorrect, the value of this feature is meaningless. So it works best not on its own but when paired with other features.

4.3.4 Classification Algorithm

To assign a label to each fixation, we use a simple feature weighting procedure:

$$\ell_t = \arg\min_i (\gamma p_t^i + (1-\gamma) v_t^i),$$

in which the label is assigned based on a weighted linear combination of the position and velocity features (represented by the parameter γ). While more sophisticated classification algorithms are possible, the natural meaningfulness of the features chosen combined with the desire to evaluate the features as directly as possible means that higher complexity is unnecessary. In addition, if we wish to extend the labeling procedure to produce a distribution over keypoints rather than a keypoint itself, we can instead use the softmax

$$p(\ell_t = i) = \text{soft}\max_i -(\gamma p_t^i + (1-\gamma) v_t^i).$$

To evaluate this algorithm, we compare three different variants. For the first algorithm to compare, we use the position features only and discard the velocity

features; that is, we set $\gamma = 1$. Next, since the feature v_t^i depends on the value of ℓ_{t-1}, we derive this previous value in two different ways. In one variant, the *true position-velocity* variant, we use the true value of ℓ_{t-1} when finding the value of ℓ_t. This variant is not representative of how this algorithm would be used in practice, as it requires access to the true label. However, it isolates the classification of each data point so that an incorrect classification does not cause later problems, and thus it more directly represents the ideal power of these features. We also evaluate the *sequential position-velocity* variant, in which we use the labeled value of ℓ_{t-1}. This method represents how this algorithm would be used in practice, but it is more susceptible to error stackup. These three approaches are compared in the following section.

4.3.5 Evaluation of Position and Velocity Features

To determine the usefulness of the velocity features described above, we perform two evaluations. For the first, we generate synthetic eye gaze data so that the gaze error can be more properly controlled. For the second, we hand-labeled the fixations for the HARMONIC dataset (see Sec. 3.2.3) and evaluated the semantic gaze labeling procedure on that data. Throughout, we use $\gamma = 0.8$, as determined through cross-validation. In both cases, we find that using velocity features as part of the classification strategy increases the robustness of the classification accuracy to larger differences between gaze and keypoint positions.

We generated a synthetic dataset to ensure that the velocity features were useful in an idealized case. To build this dataset, we first generate trajectories for four keypoints in an image frame by having them follow random Gaussian walks starting from uniformly random starting positions. Then, we generate a randomized gaze signal by first generating a random segmentation of the time period following typical eye gaze dynamics to generate fixations and then assigning each of these fixation to one of the keypoints at random. Finally, the gaze signal during each fixation is determined by adding random noise around the corresponding keypoint position. This procedure is repeated to generate 200 keypoint and gaze trajectory sets of length 33.3 seconds each (1000 samples). To simulate the effect of a constant offset, an additional error of fixed magnitude and direction randomized per trajectory is added to the gaze

signal. This simulated gaze signal is then processed according to the semantic gaze labeling procedure described above.

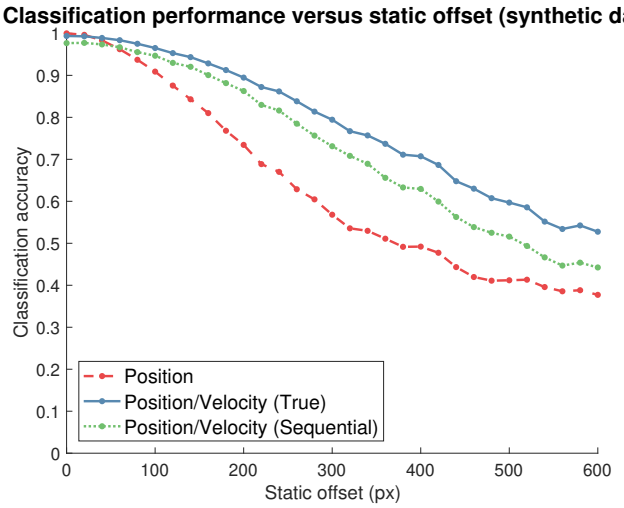


Figure 4.4: Classification accuracy on synthetic dataset. (Figure from Aronson and Admoni [2].)

The results of the algorithm on this synthetic dataset is shown in Fig. 4.4. This figure plots the overall classification accuracy on the synthetic dataset as a function of the magnitude of the offset added to the data. As expected, when the offset is very small, the position and velocity features perform similarly. However, as the magnitude of the offset increases, the velocity-based classification strategies stay more accurate, whereas the position-only strategy decreases in accuracy. Thus, the velocity features succeed at being more robust towards constant offsets. In addition, the true position-velocity strategy outperforms the sequential true position-velocity strategy, but even the sequential strategy gives benefits over the position-only strategy.

Next, we evaluate these algorithms on the HARMONIC dataset (see Sec. 3.2.3). For keypoint locations, we used a tag grid present in the egocentric video frame to compute the egocentric camera extrinsics, which we then smoothed using a Kalman filter. Then, object positions were projected into the egocentric camera using these extrinsics and a prior tag grid location calibration step.

To obtain ground-truth labels, four coders examined each fixation that occurred in the 120 teleoperation-only trials and assigned it a label corresponding to each robot joint and morsel target, or -1 if the fixation was determined to be noise. In addition,

all coders coded the same 10% of the trials (randomly selected), and the average pairwise Cohen's kappa (inter-rater reliability) score was 0.645, indicating acceptable agreement. Determining dependence on the error magnitude is more difficult than in synthetic data, as the offset is not controllable. To calculate this dependency, we first calculated the angle distance between each fixation and the mean position of its true label, then we binned fixations based on this value with a width of 0.6° and discarded bins with fewer than 20 members.

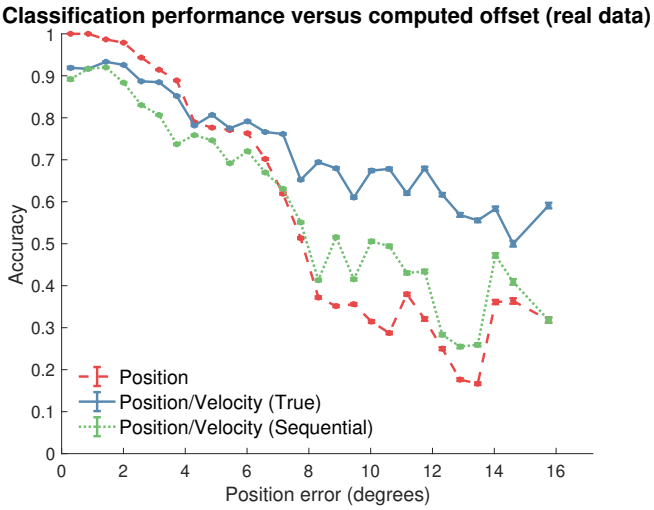


Figure 4.5: Classification accuracy on the HARMONIC dataset. (Figure from Aronson and Admoni [2].)

Fig. 4.5 shows the accuracy of our classification strategies applied on the HARMONIC dataset as a function of offset bin. As in the synthetic data, all methods have pretty good accuracy for small errors, though the position-only method slightly outperforms the others. As the offset increases, the position-only and sequential methods drop off, whereas the true velocity method maintains its performance. Thus, velocity features are indeed useful for improving the accuracy of semantic labeling.

4.4 Proposed Work: Sequence Modeling

The pipeline above describes how to process the raw gaze signal obtained by a sensor to recover a sequence of foveated scene objects (drawn from a finite set) along with

start and end ties. We propose to use this sequence as an input to various learning algorithms to make inferences about the users' mental state.

4.4.1 Goal Prediction

The most direct use of eye gaze, and one that is well-supported in the literature [72, 6, 38], is to determine a user's intended goal. In our benchmark task, users manipulated the robot to pick up one of three different morsels on the plate. Users reported which of the three morsels they were intending to spear, and their success was reported for each trial. Therefore, this task serves as a good benchmark to evaluate this signal's ability to reflect the user's goal within this context.

4.4.2 Failure Detection

Another use of this process eye gaze signal is to determine when something has gone wrong during the process. As discussed above, when something goes wrong, people's eye gaze patterns often change: they look at unexpected objects or elements of the scene that are causing failures. Therefore, this behavior suggests that this eye gaze pattern can reveal that people perceive there to be a problem with the robot's behavior using a signal almost completely separate from the robot's own self-analysis of its performance.

4.4.3 Challenges

Sensitivity to errors in sequence data. In our dataset, the three goal morsels were placed close together within the gaze space. Due to the gaze error, it can be especially difficult to distinguish between these goals. For better goal prediction results, a more reliable gaze tracking system, a more robust labeling procedure, or redesigning the experiment to ease their differentiability should lead to better results.

Handling fixation durations. The eye gaze signal output from the pipeline consists of labels paired with durations of the fixation, which can persist anywhere between 200-1200 ms. Standard hidden Markov models (HMMs), which are a common strategy to analyze sequence data and have been applied successfully to gaze signals

in the past [52] do not consider this timing. Two approaches to handle the time are to (1) neglect the time itself and consider the entire fixation as a single observation or (2) resample the sequence based on some consistent interval, repeating labels for as many samples as they contain. The first approach neglects important information and is sensitive to changes in the event detection procedure; the second can lead to data imbalance. It may be fruitful to consider extensions of HMMs and Markov models that include state durations.

Obtaining ground truth for user intention. While measuring user intent is useful, it is difficult to obtain ground-truth labels. Often, people are not conscious of their intentions or their intentions change during the process. For the HARMONIC dataset, participants reported their intended morsel before beginning the trial, but some changed which morsel they sought during the task. Furthermore, intentions such as rectifying kinematic failures or obstacle avoidance are difficult to identify even by hand, as they often do not have clear start or end points, and the user may be trying to solve that problem while simultaneously making task progress elsewhere. Therefore, to build actionable user intention models, we must design specific situations where experiments can yield meaningful results.

4.5 Conclusion

In this chapter, we laid out a pipeline for processing the raw eye gaze signal into a state usable for assistance. First, we discussed related work on signal collection and event detection. Next, we described our completed work on semantic gaze labeling. Finally, we proposed some strategies for analyzing the signal in ways that will lead to use in assistance and showed some preliminary results. In the next chapter, we discuss how to use this processed signal for manipulation assistance.

4.5.1 Completed Work

The velocity features and full pipeline were presented at the ACM Eye Tracking Research and Applications (ETRA) conference in 2019 [2].

4.5.2 Proposed Work

Ongoing work to improve this pipeline may be published at ETRA, but no specific additions are proposed as part of this thesis. Proposed uses of the gaze data are discussed in the next chapter.

Chapter 5

Eyegaze for Assistance

5.1 Introduction

Now that we have a method of transforming the eye gaze and context information into a common format, we can use that signal to apply our assistive algorithms. In this section, we present two approaches to assistance: *goal assistance*, in which gaze reveals the user's goal, and *dynamic obstacle avoidance*, in which gaze reveals a user's concern about particular workspace obstacles. Both behaviors combine inference on the labeled fixations described above with an assistance behavior that extends the shared autonomy framework [4]. These two behaviors represent distinct ways to use the mental state information that gaze reveals in explicit assistive strategies and demonstrates that eye gaze is a powerful signal for assistive feedback.

In *goal assistance*, we learn a distribution over intended user goals from users' eye gaze behavior. We can then combine this goal prediction behavior with a prediction derived from the user's control inputs. Eye gaze and user control input are complementary signals for goal inference: eye gaze gives global information (focused directly on the goal), whereas control input gives relative information (the next step on the path from the current point). Therefore, we hypothesize that combining eye gaze with control input will give a more robust goal prediction throughout the task and improve the assistance quality. While inferring goal from gaze has been explored in other efforts, this project represents the first time it is applied in an assistance framework.

In *dynamic obstacle avoidance*, we learn from the user's gaze how important it is to avoid particular objects in the workspace. In particular, we hypothesize that people will look at an obstacle when they are particularly concerned about it interfering with the robot's task and will not look at it otherwise. To use this observation, we develop an assistance behavior that has the robot avoid obstacles with a wider radius when a user looks at them more. This behavior may occur because the user knows more about the object than the robot does (the object is fragile, hot, incorrectly detected, etc) or because the user does not trust the robot's behavior and prefers a larger safety margin. In either case, the robot's increased caution is an appropriate response. This obstacle avoidance behavior is a novel usage of gaze and should pave the way for more creative uses of the signal.

We begin by summarizing the shared autonomy framework, upon which both of these behaviors are built. Then, we describe the goal inference assistance, including how the probability over goals is determined, how it is incorporated into the assistance behavior, and how we propose to validate it in a user study. Next, we describe the dynamic obstacle avoidance behavior, assistance, and proposed evaluation. Finally, we conclude by proposing additional ways the eye gaze signal might be used for assistance.

5.2 Background: Shared Autonomy Assistance

5.2.1 Overview

To add assistance, we build on *shared autonomy*, a framework for shared control introduced by Javdani et al. [4]. We briefly outline that approach here; for more details and evaluation, see that paper. Shared autonomy is designed to build on prior systems by generating assistance proactively whenever it can, in contrast to other models that require the robot's certainty to reach a particular threshold before providing assistance. In addition, the structure of the shared autonomy framework makes augmenting it with additional behaviors relatively straightforward.

We model the robot action planning as a Markov decision process (MDP), which is a tuple (X, A, T, C_g), with X the set of all robot states (joint position, velocity, etc.), A the set of possible robot actions (applied twists), and $T : X \times A \to X$ is the

transition function. The cost function C_g describes the robot's ideal behavior for reaching goal g, which we define here as a goal position for the robot. Given a goal g, the optimal policy $\pi_g : X \to A$ can be found using standard reinforcement learning methods.

To incorporate uncertainty over user goals, the model is extended into a partially-observable Markov decision process (POMDP) defined as the tuple $(S, A, T, C^{rob}, O, \Omega)$. $S = X \times G$ represents the robot state augmented with the user's true goal $g \in G$. A and T are the action set and transition function induced from the underlying MDP. $C^{rob} : S \times A \times O \to \mathbb{R}$ is a cost function that models how the robot incorporates both information about the goal and the observation to determine its behavior; note that in this model, the observation o is passed into the cost function. O represents the observations for the POMDP, which in this formulation is the motion command given by a user through their control interface (here, a joystick). Finally, Ω represents an observation model to determine the probability distribution ("belief") over goals $b(g)$ from the observation o.

5.2.2 User Input Observation Model

The original shared autonomy framework derives its belief exclusively using the user's input command u; that is, $O = U$. To derive this belief, the framework uses a maximum-entropy inverse reinforcement learning (MaxEnt IRL) model [87], in which users are assumed to be noisily optimizing an MDP induced by their goal. Specifically, let (X, U, T, C_g^{usr}) represent the robot control MDP described above, except that the user provides an input U (e.g., a joystick signal) which is transformed by the controller using a function $D : U \to A$. Given this model and a particular goal, assume that the user's policy follows the distribution $\pi^{usr}(u|x, g)$. To make this approach tractable, replace the traditional Bellman equation with a softmin version:

$$Q^{\approx}_{g,t}(x, u) = C_g^{usr}(x, u) + V^{\approx}_{g,t+1}(x')$$
$$V^{\approx}_{g,t}(x) = \operatorname*{soft\,min}_u Q^{\approx}_{g,t}(x, u),$$

where $\operatorname{soft\,min}(x) = -\log \int_x -\exp(f(x))\,\mathrm{d}x$ and x' is the result of applying action u in state x, i.e., $x' = T(x, D(u))$. In this formulation, $Q^{\approx}_{g,t}(x, u)$ and $V^{\approx}_{g,t}(x)$ can be

solved tractably using dynamic programming, and the policy $\pi^{usr}(x|u,g)$ is defined as

$$\pi^{usr}(x|u,g) = Q^{\approx}_{g}(x,u) - Q^{\approx}_{g}(x,0).$$

This expression differs from the expression given in Javdani et al. [4], which uses the policy $\pi^{usr}(x|u,g) = Q^{\approx}_{g,t}(x,u) - V^{\approx}_{g,t}(x)$. The original approach may give a different probability for each goal even when the user provides no input, i.e. $u = 0$. One could argue that providing no input represents implicit satisfaction with the robot's behavior, in which case using the original update rule is appropriate. However, in our experience, people are more likely to provide no input for other reasons, such as distraction, planning, mode switching, etc. Therefore, we modify the distribution to normalize against the assumption of $u = 0$ rather than the value of the current state $V^{\approx}_{g,t}(x)$. See Newman et al. [69] for additional details.

The likelihood of a given input sequence $(u_0, \cdots, u_t)$ given a particular goal g and sequence of states $(x_0, \cdots, x_t)$ is given by

$$p(u_0, \cdots, u_t|g; x_0, \cdots, x_t) = \prod_t \pi^{usr}_t(u_t|x_t, g), \tag{5.1}$$

and the goal probabilities can be computed according to Bayes' rule as

$$p(g|u_0, \cdots, u_t) = \frac{p(u_0, \cdots, u_t|g)p(g)}{\sum_{g'} p(u_0, \cdots, u_t|g')p(g')}.$$

This approach represents the observation model, which maps from the user joystick input u to a distribution over goals $p(g|u)$.

5.2.3 POMDP Solution

Now that the observation model has been defined, we turn to solving the overall POMDP. To make this behavior tractable, we make the hindsight optimization/QMDP assumption, which computes a policy assuming the uncertainty will be resolved at the next timestep. Specifically, this formulation sets

$$Q(b,a,u) = \sum_g b(g) Q_g(x,a,u),$$

the expectation of Q over the goal belief. We further make the assumption when solving the POMDP that the user will apply no further action after the current state. That is, the state-action value function Q_g is computed from the cost function $C^{rob}((x, g), a, u)$ by assuming that $u = 0$. This assumption allows the underlying MDP to be pre-computed up to the known goal rather than requiring a new solution whenever the user's input is given.

We use the cost function

$$C_g^{usr}(x, u) = \begin{cases} \alpha, & |x - x_g| > \delta \\ \frac{|x-x_g|}{\delta}\alpha, & d \leq \delta \end{cases}, \tag{5.2}$$

which attains a constant value α outside a radius δ around the goal position x_g and decreases linearly to 0 within that radius. The robot's overall cost function is then set as

$$C_g^{rob}(x, a, u) = C_g^{usr}(x, u) + (a - D(u))^2,$$

which is the original, goal-centered cost $C_g^{usr}(x, u)$ plus a quadratic penalty for the induced action a differing from the user's command u (transformed appropriately).

5.2.4 Assistance Behavior

In our model eating scenario, the shared autonomy behavior works as follows. At the beginning of the task, the robot is positioned far from all goal objects. Even with an initial uniform belief $b(g) = 1/|G|$, the POMDP can still find an action that makes progress towards the goal in expectation, since all goals require reorienting the fork above the marshmallow and moving towards the spearing plane. As users provide input, the robot updates its belief, which leads it to continue to move towards the spearing plane but also start moving towards the appropriate goal. When the robot has reached the spearing plane, assistance slows down. Since the robot is already close to its goal, the user's actions themselves are largely the same as what the assistance would provide. Once they learned this tendency, many users decided to split the assistance into two phases: they allow the robot to autonomously plan to the spearing plane to a point between all of the goals, and then they move the robot within that plane to achieve their particular goal. In the original study [4], shared autonomy was

shown to enable users to complete their tasks faster with less user input.

5.3 Proposed Work: Goal Inference

5.3.1 Inferring Goal From Gaze

Inferring people's goal from gaze traces is a classification problem that can be learned from data. The input to the problem is a sequence of timed fixations with labels matching the objects in the scene (as described in the previous chapter). The output is a probability distribution $p(f_0, \cdots f_t|g)$ modeling the likelihood that a user having a particular goal would emit the observed gaze history to this point. In addition, the methods below must have a strategy to incorporate the varying duration of the fixations into the model.

Several modeling techniques are available to solve the problem. We sketch some here, but it is ongoing work to determine the best method.

Constant goal emissions. The simplest method is to treat each fixation as an independent draw from the possible labels. Specifically, assume that the probability of any given fixation being goal-directed is fixed. That is,

$$p(f_i|g_k) = \begin{cases} \left(\frac{\alpha}{1+\alpha}\right)^{N(d_i)}, & \ell_i = g_k \\ \left(\frac{1}{1+\alpha}\right)^{N(d_i)}, & \ell_i \neq g_k \end{cases}$$

where $f_i = (\ell_i, d_i)$ represents a labeled fixation along with its duration, $\alpha > 1$ is a scale parameter, and N is a function representing how to handle the fixation duration. If $N(d) = 1$ is a constant function, the above model treats each fixation as a single draw from the goals. If $N(x) = \lfloor d_i/\Delta\tau \rfloor$ for some constant step size $\Delta\tau$, we weight the fixation by its number of samples, treating each timestep within the fixation implicitly as a separate draw from among the goals. Using this observation model, the goal probabilities become

$$p(g_k|f_0, \cdots, f_t) = \frac{\alpha^{N(k)}}{\sum_i \alpha^{N(i)}},$$

where

$$N(k) = \sum_i \delta(\ell_i = g_k) N(d_k)$$

is the total weight of all samples whose fixation label corresponds to that goal.

Contextual glances with HMMs. A more sophisticated method is to train hidden Markov models corresponding to each of the goals. This method has the advantage that it can capture some details of task context. For example, glances from the end-effector to a goal may be more meaningful than glances from an internal joint to the goal, as the first may be a monitoring glance while the second an idle glance away while working out a problem. To build this model, we can train one HMM for each goal on the HARMONIC data collected (which include annotations of the user's target goal) and use cross-validation to select the number of hidden states. For the sequence, we can use the raw labels themselves $(\ell_0, \cdots, \ell_t)$, or duplicate labels according to the fixation durations to generate a sequence

$$(\underbrace{\ell_0, \cdots, \ell_0}_{\lfloor \frac{d_0}{\Delta\tau} \rfloor}, \underbrace{\ell_1, \cdots, \ell_1}_{\lfloor \frac{d_1}{\Delta\tau} \rfloor}, \ell_2, \cdots, \ell_t).$$

Here, $\Delta\tau$ is a parameter that sets the level of quantization of the states. Then, the observation model $p(f_0, \cdots, f_t | g_k)$ is given as the probability of that fixation sequence computed by the model corresponding to g_k, and goal probabilities are found through marginalization.

Additional methods. There are several options for improving the above algorithms. One option involves more explicitly taking the duration of the fixations into account, possibly as an additional numeric feature alongside the categorical labels. A second approach involves using contextual information from the robot behavior to modify the states. Proposed work involves comparing and exploring additional algorithms.

5.3.2 Eye Gaze Goal Predictions in Shared Autonomy

To incorporate this goal prediction framework into shared autonomy, we augment the original POMDP with the fixation observation F^t, where $f \in F = (\ell_i, d_i)$ represents the sequence of fixation labels and durations described above; then the full observation O is given as $O = U \times F^t$. We make the assumption that gaze and user signal are independent conditioned on the user's goal; then, we can treat the two observation models separately and combine them in a straightforward way.

For the gaze, we require a fixation likelihood model in the form

$$p(f_0, \cdots, f_t|g),$$

which is derived according to the models described above. When using gaze only, goal probabilities are derived using a Bayesian incremental update (akin to Eqn. 5.2.2):

$$p(g|f_0, \cdots, f_t) = \frac{p(f_0, \cdots, f_t|g)p(g)}{\sum_{g'} p(f_0, \cdots, f_t|g')p(g')}. \tag{5.3}$$

To combine gaze and control predictions, the conditional independence assumption allows us to predict goal probabilities from each observation separately and combine them at the end according to the equation

$$p(g|f_0, \cdots, f_t; u_0, \cdots, u_t) = \frac{p(g|f_0, \cdots, f_t)p(g|u_0, \cdots, u_t)p(g)}{\sum_{g'} p(g'|f_0, \cdots, f_t)p(g'|u_0, \cdots, u_t)p(g')},$$

where $p(g|f_0, \cdots, f_t)$ and $p(g|u_0, \cdots, u_t)$ are computed according to the observation models presented in Eqn. 5.3 and Eqn. 5.1 respectively.

5.3.3 Goal Assistance User Study

To evaluate this procedure, we intend to run a user study to compare the relative benefits of joystick-based and gaze-based goal inference for assistance. We expect to evaluate the following claims:

H1. Eye gaze provides goal information earlier in the task than joystick does.

H2. Eye gaze and joystick together provide better goal predictions earlier than either do alone.

H3. People prefer assistance based on both eye gaze and joystick than using only one or the other.

The user study will be similar to the spearing study described in Sec. 3.2.2, using the same object spearing task as described there. However, we propose to alter the conditions to {no assistance, assistance from control input only, assistance from gaze only, assistance from combined control and gaze}. We anticipate running a similar within-subjects fully counterbalanced study with non-disabled participants. To validate **H1** and **H2**, we can calculate the integral of log loss of goal prediction accuracy versus progress through the task among each conditions. For **H3**, we will ask participants to evaluate each condition and rank the conditions similar to the measures used in Javdani et al. [4].

5.4 Proposed Work: Responsive Obstacle Avoidance

Using eye gaze to understand people's goals is a promising strategy for improving assistance. However, the eye gaze signal, as we have determined, reveals even more about the task than just people's intended goal. We focus specifically on one aspect of mental state: people's concern about navigating around a specific obstacle in the workspace. We pair this aspect with an assistance method that adapts the robot's path to avoid important objects more widely.

One of our insights from analyzing people's eye gaze behavior is that people look at areas of the scene that are likely to cause them problems. As described above, we identified several examples of people looking at a joint of the robot that was in a kinematically problematic configuration. However, during normal operation people almost never look at the internal joints of the robot. In addition, work on by-hand manipulation [6] suggests that people look at obstacles they are likely to encounter when navigating around them, and don't look at them otherwise. Therefore, we hypothesize that if an obstacle is present in the workplace, people will look at it while they believe the robot needs to actively avoid it, and will not look at it otherwise.

This behavior induces a new strategy for assistance. In particular, by monitoring when users look at specific workspace obstacles, an assistance system can determine

when the user believes those obstacles to be particularly problematic in the path. From this knowledge, the robotic system can then change how it responds to the obstacle. If the user expresses concern (by looking more than usual at the obstacle), the robot can give it a wider berth. If the user does not seem concerned by the obstacle (by looking elsewhere), the robot can take a more efficient path around it. (This distinction calls back to the tradeoff between *predictable* and *legible* paths [88].)

The impulse for a user to be concerned about a particular obstacle can be caused by multiple things. If we assume that the robot has a perfect environment model and safely navigates around objects, a user might still be unfamiliar with the robot's operation and be reluctant to trust it. Even if the robot is operating in a safe behavior, avoiding the obstacle more clearly can help the user acquire trust in the robot's operation. Alternatively, the user may know more about the environment than the robot does. The robotic system may have an error in its object localization, or the object may have non-obvious traits (like fragility or heat) that the robot does not know how to detect. Then, the robot giving the obstacle a wider berth will result in safer performance. Importantly, the system does not need to know which of these incidents is the case. The robot avoiding the object will produce useful behavior whatever the reason the user has for provoking it.

5.4.1 Detecting Obstacles to Avoid

The first step in enabling this assistance behavior is to determine when a user is particularly concerned about an obstacle. As noted above, we have evidence that people look at a region of the scene that is causing them problems. Furthermore, research on by-hand manipulation [6] suggest that people look at obstacles when they are problems, and ignore them otherwise. Therefore, we expect that we can use an object-oriented glance detection framework to identify when people are concerned about a particular obstacle.

Specifically, we need to determine a score function $r_{obs}((f_0, \cdots, f_t))$ that, given a sequence of fixations and an object label *obs*, determines how much attention people are paying to that particular object. We will evaluate several possibilities for this function, including moving average of fixations to the object versus other fixations or a learned function on a dataset hand-labeled with times when the obstacle is important.

Functions will be evaluated through piloting and testing on the HARMONIC dataset.

5.4.2 Obstacle Avoidance Algorithm

Once we have a measure for how important an obstacle is, we need to update the robot's assistance accordingly. We propose to modify the cost function of the underlying reinforcement learning framework to change the robot's behavior. As described above in Sec. 5.2, we use a POMDP for assistance planning combined with MaxEnt IR for goal inference. The cost function we use (Eqn. 5.2) contains a constant penalty α outside a radius δ from the goal which linearly decreases to 0 inside the radius. To add obstacle avoidance, we can add a second term to the user cost function:

$$C_{obs}^{usr}(x,y) = \begin{cases} 0, & d > \Delta \\ \left(1 - \frac{d}{\Delta}\right)\beta, & d \leq \Delta \end{cases},$$

which is a linear penalty for approaching too close to an obstacle defined by the margin Δ and the scale β. Then, the total cost function is

$$C_g^{usr} = C_g^{usr}(x,u) + C_{obs}^{usr}(x,y),$$

the sum of the above cost functions.

We can modify the relative importance of the obstacle (and therefore the avoidance distance) by modifying the scaling factor β. When this value is larger, the robot will settle to a larger obstacle distance (within the radius Δ). When β is smaller, the robot is penalized less for going closer to it when pursuing its intended goal. Therefore, we can perform an update at each time step

$$\beta_{t+1} = \beta_t * \gamma^{r_t},$$

where γ is an update factor and r_t is our concern function above. When $r_t > 0$, indicating positive concern, β increases; when $r_t \leq 0$, indicating lack of concern, β decreases.

For this project, we will fill out this reinforcement learning framework by adapting it to run in real time and to match the intent prediction used by the MaxEnt IRL

algorithm. Furthermore, we will explore alternate formulations, such as potential fields, that may lead to more successful implementations of this assistance.

5.4.3 Obstacle Avoidance User Study

To evaluate our obstacle avoidance assistance mode, we will again conduct a user study. We plan to perform a similar study to the spearing one described above (Sec. 3.2.2), but add an obstacle in the workplace that the user must avoid. For study conditions, we will consider: (1) No assistance, in which the user teleoperates the robot directly. (2) Unchanging assistance, which uses the shared autonomy algorithm with obstacle avoidance but no dynamic updates. (3) Gaze-independent dynamic updates, where we use our obstacle avoidance update plan but design the concern function $r(t)$ to include only gaze-independent metrics, such as measuring the distance between the robot end-effector and the obstacle. (4) Gaze-based obstacle avoidance, as described here. Furthermore, we can have two conditions for the obstacle: (1) stationary but safe obstacle, (2) precarious obstacle (e.g. balance an object on top of our obstacle). To evaluate the study, we will measure task completion time and user satisfaction with the robot's behavior.

5.5 Conclusion

In this chapter, we propose two uses of eye gaze to improve automated assistance. In the first, *goal assistance*, we use the semantic gaze labeling signal to infer the user's goal and adapt the shared autonomy framework to include measurement of this signal. While we have made progress in developing and evaluating this system, these results require additional exploration and evaluation. In the second, *responsive obstacle avoidance*, we propose a new strategy of using eye gaze: recognizing when the user expects to be especially cautious about maneuvering around an obstacle and adapting the assistance to match. We have begun to explore this application, but additional work remains to develop the assistance algorithm and evaluate it in a user study.

5.5.1 Completed Work

We have conducted preliminary work on predicting user goal from their fixation sequence, which has been presented at the Robotics Institute Summer Scholars program.

5.5.2 Proposed Work

In this section, we propose two projects which will result in up to three papers. First, the goal prediction model and study will be submitted to HRI 2021 in October 2020. Second, the gaze responsiveness model and update method will be submitted to a conference such as RSS in early 2021, and a user study to evaluate dynamic obstacle avoidance for assistance will be submitted to HRI 2022 in October 2021.

Chapter 6

Conclusions

6.1 Summary

In this proposal, we describe how to use natural eye gaze to improve automated assistance during shared control of a robot arm. We describe multiple studies of people's eye gaze behavior while teleoperating a robot and found that people display consistent gaze patterns as they look at their goals and at potential problems in their task. We develop a pipeline for incorporating contextual information about the task into the raw eye gaze sensor data and propose several machine learning approaches to translating that data into information of the operator's mental state. Finally, we propose two gaze-based assistance approaches. First, we proposed to modify goal-based assistance by incorporating gaze-based goal predictions and comparing how well gaze predicts users' goals relative to approaches that use only their control signals.

Beyond these eye gaze results, moreover, this work shows how observing people's natural nonverbal behaviors can enhance human-robot collaboration approaches. People continuously and passively telegraph their intentions through their eyes, their body posture, their gestures, their stance, and many other ways, and other people can read their intentions and concerns directly and implicitly from these signals. Accessing these signals will enable robots to collaborate more fluently with their human partners by anticipating their needs. Moreover, the sophistication of this signal enables more complex assistance paradigms, such as adapting dynamically to

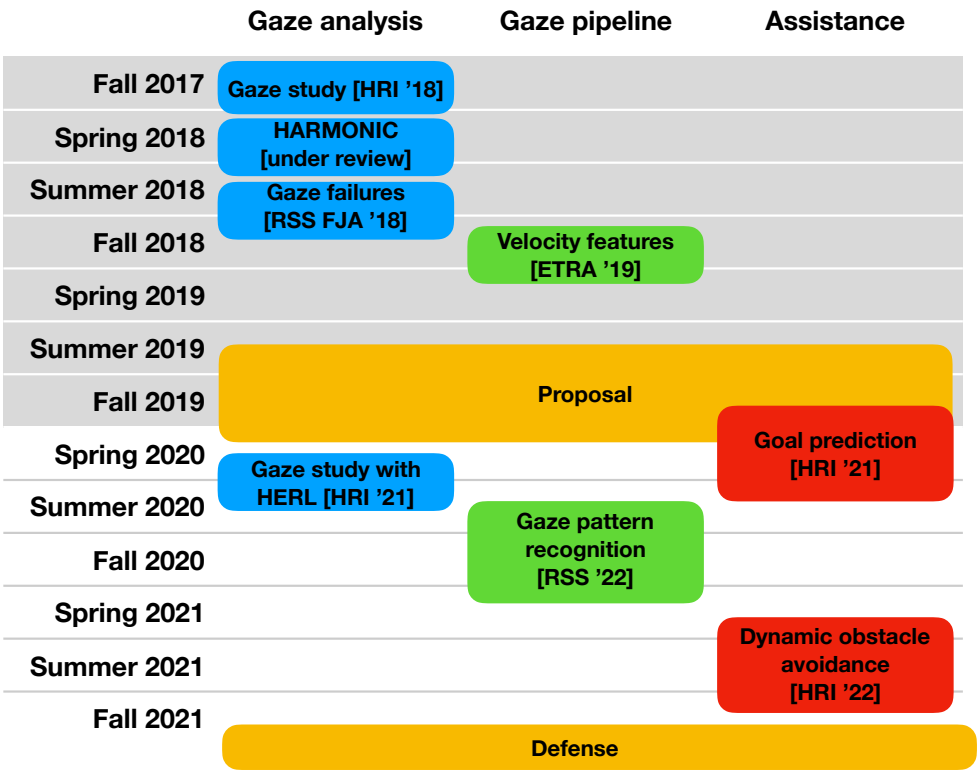
the user's preferences about how a task ought to be performed or implicitly learning that an interaction should be performed in a particular way from the user's behavior. This thesis lays the groundwork for broad investigation of the use of natural behaviors for sophisticated robotic assistance during shared control.

6.2 Future Work

Other work in the eye gaze community has found many different aspects of mental state that can be detected from eye gaze signals alone. These aspects include a user's expertise in completing a task [11, 89, 10], which task a user is performing [80, 44], or even the operator's cognitive load [12]. Furthermore, the analyses conducted here can be extended to predict an operator's next step during a multi-stage task or when they are paying attention to the task [8, 9]. Building models to learn users' intentions or states based on these signals and developing additional assistance paradigms is a natural extension of our work.

Moreover, work remains to translate this assistive paradigm directly to the assistive devices that inspire it. To validate that these assistance approaches are indeed useful for people with upper mobility impairments, we must evaluate them in a study with participants who would use a wheelchair-mounted robot arm in daily life as an assistive device. Eventually, we hope to make this type of assistance available for commercial assistive robot arms.

6.3 Timeline of Proposed Work



Appendix A

3D Math

A.1 Introduction

While Sec. 4.3 describes the general construction of the features used for the semantic gaze labeling problem, care must be taken when computing actual feature values. Specifically, we attain better accuracy modeling gaze and keypoint directions as rays from the camera center instead of 2D elements in an image frame (see Fig. A.1). In this appendix, we detail the calculations required for this approach.

A.2 Parsing gaze values

Gaze locations are typically given as ordered pairs (u_τ, v_τ) representing pixel locations in an egocentric camera corresponding to where the user is looking. While all the math described above can be done directly using these pixel locations and Euclidean distances, that method introduces projective distortion when the user is looking further from the focal center of the egocentric camera. Therefore, to remove distortion from these values, we must perform calculations using 3D rotations.

First, we must convert the gaze location (u_τ, v_τ) to a ray in 3D from the camera's

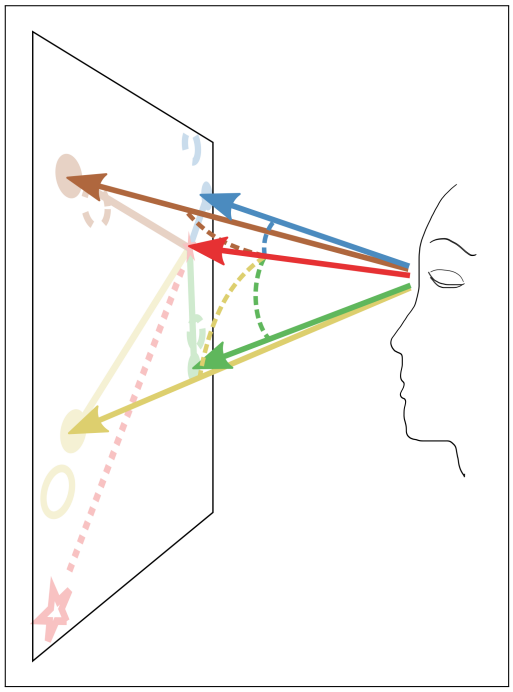


Figure A.1: In contrast to the 2D semantic gaze labeling procedure shown in Fig. 4.3, we compute all features in a vectorized, 3D model. (Figure from Aronson and Admoni [2].)

origin. To do so, we first compute a projective representation of the gaze ray,

$$\begin{bmatrix} x_\tau \\ y_\tau \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u_\tau \\ v_\tau \\ 1 \end{bmatrix}, \tag{A.1}$$

where K is the intrinsic matrix for the egocentric camera, calculated using camera calibration. Using the same equation, we can transform the keypoint pixel locations (u^i_τ, v^i_τ) to their equivalent projective ray (kx^i_τ, ky^i_τ). For simplicity, let $f_t = (x_t, y_t, 1)$ be the fixation mean location represented in projective coordinates, and $k^i_t = (kx^i_\tau, ky^i_\tau, 1)$ be the keypoint location for fixation t in projective coordinates.

A.3 Scalar position distance

To compute the position features, we must have a (scalar) distance function d_p over the projective vectors. We can use the cosine distance function, defined as

$$d_p(f_t, k_t^i) = |\frac{f_t \cdot k_t^i}{||f_t||||k_t^i||}|,$$

which represents the absolute value of the cosine of the angle between the two vectors. This representation removes any distance component and considers only the angle between the gaze direction and the keypoint and is conveniently bounded between 0 and 1.

A.4 Vector position distance

To compute velocity features, we need to have a vectorized notion of the distance between two projective representations. Here, we use the rotation between the two rays, i.e., the 3D rotation that would move one ray onto the other. We use a quaternion representation for 3D rotations.

To compute this quaternion, we must compute the axis and the angle of rotation. The axis of rotation is the cross product of the two vectors,

$$\hat{n}_t = f_{t-1} \times f_t,$$

and the angle between the two is

$$\theta_t = \arccos(f_{t-1} \cdot f_t).$$

Finally, we put them together to get the quaternion representation,

$$\overrightarrow{d_p}(f_{t-1}, f_t) = \cos\frac{\theta_t}{2} + \sin\frac{\theta_t}{2}(\hat{n}_{t,1}\hat{i} + \hat{n}_{t,2}\hat{j} + \hat{n}_{t,3}\hat{k}).$$

The same calculation gives the rotation between successive keypoints as required by the velocity feature calculation.

A.5 Velocity distance

To compare velocity features, we need a metric over velocities, expressed as quaternions. One standard metric to use is

$$d_v(p, q) = 1 - (p \cdot q)^2,$$

which is a function of the cosine of the angle of the rotation equivalent to composing one rotation with the inverse of the second. This metric is also, conveniently, restricted to the range $[0, 1]$.

Bibliography

[1] Reuben M. Aronson, Thiago Santini, Thomas. C. Kübler, Enkelejda Kasneci, Siddhartha Srinivasa, and Henny Admoni. Eye-Hand Behavior in Human-Robot Shared Manipulation. In *ACM/IEEE International Conference on Human-Robot Interaction*, 2018.

[2] Reuben M. Aronson and Henny Admoni. Semantic gaze labeling for human-robot shared manipulation. In *Eye Tracking Research and Applications Symposium (ETRA)*. Association for Computing Machinery, 6 2019. ISBN 9781450367097. doi: 10.1145/3314111.3319840.

[3] Reuben M. Aronson and Henny Admoni. Gaze for Error Detection During Human-Robot Shared Manipulation. In *Fundamentals of Joint Action workshop, Robotics: Science and Systems*, 2018.

[4] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and collaboration. *The International Journal of Robotics Research*, 2018.

[5] Deepak Gopinath, Siddarth Jain, and Brenna D. Argall. Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE Robotics and Automation Letters*, 2(1):247–254, 1 2017. ISSN 23773766. doi: 10.1109/LRA.2016.2593928. URL http://ieeexplore.ieee.org/document/7518989/.

[6] Roland S Johansson, Gö Ran Westling, Anders Bäckström, and J Randall Flanagan. Eye–Hand Coordination in Object Manipulation. *The Journal of Neuroscience*, 21(17):6917–6932, 2001.

[7] J Randall Flanagan, Miles C Bowman, and Roland S Johansson. Control strategies in object manipulation tasks, 12 2006. ISSN 09594388. URL http://linkinghub.elsevier.com/retrieve/pii/S0959438806001450.

[8] Michael Land, Neil Mennie, and Jennifer Rusted. The Roles of Vision and Eye Movements in the Control of Activities of Daily Living. *Perception*, 28 (11):1311–1328, 11 1999. ISSN 0301-0066. doi: 10.1068/p2935. URL http://journals.sagepub.com/doi/10.1068/p2935.

[9] Mary M. Hayhoe, Anurag Shrivastava, Ryan Mruczek, and Jeff B. Pelz. Visual memory and motor planning in a natural task. *Journal of Vision*, 3(1):6, 2 2003. ISSN 1534-7362. doi: 10.1167/3.1.6. URL http://jov.arvojournals.org/article.aspx?doi=10.1167/3.1.6.

[10] Yan Liu, Pei Yun Hsueh, Jennifer Lai, Mirweis Sangin, Marc Antoine Nüssli, and Pierre Dillenbourg. Who is the expert? Analyzing gaze data to predict expertise level in collaborative applications. In *Proceedings - 2009 IEEE International Conference on Multimedia and Expo, ICME 2009*, pages 898–901, 2009. ISBN 9781424442911. doi: 10.1109/ICME.2009.5202640.

[11] Uta Sailer, J. Randall Flanagan, and Roland S. Johansson. Eye–Hand Coordination during Learning of a Novel Visuomotor Task. *Journal of Neuroscience*, 25 (39), 2005. URL http://www.jneurosci.org/content/25/39/8833.long.

[12] Jackson Beatty. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological Bulletin*, 91(2):276–292, 3 1982. ISSN 00332909. doi: 10.1037/0033-2909.91.2.276.

[13] Carol A. Stanger, Carolyn Anglin, William S. Harwin, and Douglas P. Romilly. Devices for Assisting Manipulation: A Summary of User Task Priorities. *IEEE Transactions on Rehabilitation Engineering*, 2(4):256–265, 1994. ISSN 10636528. doi: 10.1109/86.340872. URL http://ieeexplore.ieee.org/document/340872/.

[14] William S. Harwin, Tariq Rahman, and Richard A. Foulds. A Review of Design Issues in Rehabilitation Robotics with Reference to North American Research. *IEEE Transactions on Rehabilitation Engineering*, 3(1):3–13, 1995. ISSN 10636528. doi: 10.1109/86.372887.

[15] Cheng Shiu Chung, Hongwu Wang, and Rory A. Cooper. Functional assessment and performance evaluation for assistive robotic manipulators: Literature review, 7 2013. ISSN 10790268.

[16] Gert Willem R.B.E. Römer, Harry J.A. Stuyt, and Albér Peters. Cost-savings and economic benefits due to the Assistive Robotic Manipulator (ARM). In *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics*, volume 2005, pages 201–204, 2005. ISBN 0780390032. doi: 10.1109/ICORR.2005.1501084.

[17] Veronique Maheu, Philippe S. Archambault, Julie Frappier, and François Routhier. Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In *IEEE International Conference on Rehabilitation Robotics*, 2011. ISBN 9781424498628. doi: 10.1109/ICORR.2011.5975397.

[18] Laura V. Herlant, Rachel M. Holladay, and Siddhartha S. Srinivasa. Assistive

teleoperation of robot arms via automatic time-optimal mode switching. In *ACM/IEEE International Conference on Human-Robot Interaction*, volume 2016-April, pages 35–42. IEEE, 3 2016. ISBN 9781467383707. doi: 10.1109/HRI.2016.7451731. URL http://ieeexplore.ieee.org/document/7451731/.

[19] Dana H. Ballard and Mary M. Hayhoe. Modelling the role of task in the control of gaze. *Visual Cognition*, 17(6-7):1185–1204, 8 2009. ISSN 13506285. doi: 10.1080/13506280902978477. URL http://www.tandfonline.com/doi/abs/10.1080/13506280902978477.

[20] Benjamin W. Tatler, Mary M. Hayhoe, Michael F. Land, and Dana H. Ballard. Eye guidance in natural vision: reinterpreting salience., 5 2011. ISSN 15347362. URL http://jov.arvojournals.org/Article.aspx?doi=10.1167/11.5.5.

[21] Marco Ramacciotti, Mario Milazzo, Fabio Leoni, Stefano Roccella, and Cesare Stefanini. A novel shared control algorithm for industrial robots. *International Journal of Advanced Robotic Systems*, 13(6):1–10, 12 2016. ISSN 17298814. doi: 10.1177/1729881416682701. URL http://journals.sagepub.com/doi/10.1177/1729881416682701.

[22] Dinh Son Vu, Ulysse Côté Allard, Clément Gosselin, François Routhier, Benoit Gosselin, and Alexandre Campeau-Lecours. Intuitive adaptive orientation control of assistive robots for people living with upper limb disabilities. In *IEEE International Conference on Rehabilitation Robotics*, pages 795–800. IEEE Computer Society, 8 2017. ISBN 9781538622964. doi: 10.1109/ICORR.2017.8009345.

[23] Panadda Marayong, Ming Li, Allison M. Okamura, and Gregory D. Hager. Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2, pages 1954–1959, 2003. doi: 10.1109/robot.2003.1241880.

[24] Jacob W. Crandall and Michael A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 1290–1295, 2002. doi: 10.1109/irds.2002.1043932.

[25] Erkang You and Kris Hauser. Assisted teleoperation strategies for aggressively controlling a robot arm with 2D input. In *Robotics: Science and Systems*, volume 7, pages 354–361, 2012. ISBN 9780262517799. doi: 10.7551/mitpress/9481.003.0050.

[26] Daniel Aarno and Danica Kragic. Motion intention recognition in robot assisted applications. *Robotics and Autonomous Systems*, 56(8):692–705, 8 2008. ISSN 09218890. doi: 10.1016/j.robot.2007.11.005. URL http://www.sciencedirect.com/science/article/pii/S0921889007001704.

[27] Kris Hauser. Recognition, prediction, and planning for assisted teleoperation of

freeform tasks. In *Autonomous Robots*, volume 35, pages 241–254. Springer US, 11 2013. doi: 10.1007/s10514-013-9350-3. URL http://link.springer.com/10.1007/s10514-013-9350-3.

[28] Eric Demeester, Alexander Hüntemann, Dirk Vanhooydonck, Gerolf Vanacker, Alexandra Degeest, Hendrik Van Brussel, and Marnix Nuttin. Bayesian estimation of wheelchair driver intents: Modeling intents as geometric paths tracked by the driver. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5775–5780, 2006. ISBN 142440259X. doi: 10.1109/IROS.2006.282386.

[29] Alexander Hüntemann, Eric Demeester, Gerolf Vanacker, Dirk Vanhooydonck, Johan Philips, Hendrik Van Brussel, and Marnix Nuttin. Bayesian plan recognition and shared control under uncertainty: Assisting wheelchair drivers by tracking fine motion paths. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3360–3366, 2007. ISBN 1424409128. doi: 10.1109/IROS.2007.4399524.

[30] Alexander Hüntemann, Eric Demeester, Marnix Nuttin, and Hendrik Van Brussel. Online user modeling with Gaussian Processes for Bayesian plan recognition during power-wheelchair steering. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 285–292, 2008. ISBN 9781424420582. doi: 10.1109/IROS.2008.4651040.

[31] Tom Carlson and Yiannis Demiris. Collaborative Control in Human Wheelchair Interaction Reduces the Need for Dexterity in Precise Manoeuvres. Technical report.

[32] Anca D Dragan and Siddhartha S Srinivasa. A policy-blending formalism for shared control. In *International Journal of Robotics Research*, volume 32, pages 790–805. SAGE PublicationsSage UK: London, England, 6 2013. doi: 10.1177/0278364913490324. URL http://journals.sagepub.com/doi/10.1177/0278364913490324.

[33] Katharina Muelling, Arun Venkatraman, Jean Sebastien Valois, John E Downey, Jeffrey Weiss, Shervin Javdani, Martial Hebert, Andrew B Schwartz, Jennifer L Collinger, and J. Andrew Bagnell. Autonomy infused teleoperation with application to brain computer interface controlled manipulation. *Autonomous Robots*, 41(6):1401–1422, 2017. ISSN 15737527. doi: 10.1007/s10514-017-9622-4.

[34] Deepak E Gopinath and Brenna D Argall. Mode switch assistance to maximize human intent disambiguation. In *Robotics: Science and Systems*, volume 13, 2017. ISBN 9780992374730. doi: 10.15607/rss.2017.xiii.046. URL http://www.roboticsproceedings.org/rss13/p46.pdf.

[35] Kristian Lukander, Miika Toivanen, and Kai Puolamäki. Inferring intent and action from gaze in naturalistic behavior: A review, 10 2017. ISSN 19423918.

[36] Thomas Bader, Matthias Vogelgesang, and Edmund Klaus. Multimodal integration of natural gaze behavior for intention recognition during object manipulation. In *ICMI-MLMI'09 - Proceedings of the International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interfaces*, pages 199–206, 2009. ISBN 9781605587721. doi: 10.1145/1647314.1647350.

[37] Ayaka Matsuzaka, Liu Yang, Chuangyu Guo, Takuya Shirato, and Akio Namiki. Assistance for Master-Slave System for Objects of Various Shapes by Eye Gaze Tracking and Motion Prediction. In *2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018*, pages 1953–1958. IEEE, 12 2018. ISBN 9781728103761. doi: 10.1109/ROBIO.2018.8664898. URL https://ieeexplore.ieee.org/document/8664898/.

[38] Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 83–90, 2016.

[39] Nuno Ferreira Duarte, Mirko Rakovic, Jovica Tasevski, Moreno Ignazio Coco, Aude Billard, and Jose Santos-Victor. Action Anticipation: Reading the Intentions of Humans and Robots. *IEEE Robotics and Automation Letters*, 3 (4):4132–4139, 2018. ISSN 23773766. doi: 10.1109/LRA.2018.2861569. URL https://arxiv.org/pdf/1802.02788.pdf.

[40] Ali Borji and Laurent Itti. Defending yarbus: Eye movements reveal observers' task. *Journal of Vision*, 14(3), 2014. ISSN 15347362. doi: 10.1167/14.3.29.

[41] Weilie Yi and Dana Ballard. Recognizing behavior in hand-eye coordination patterns. *International Journal of Humanoid Robotics*, 6(3):337–359, 2009. doi: 10.1142/S0219843609001863. URL http://www.ncbi.nlm.nih.gov/pubmed/20862267.

[42] Peter Hevesi, Jamie A Ward, Orkhan Amiraslanov, Gerald Pirkl, and Paul Lukowicz. Analysis of the Usefulness of Mobile Eyetracker for the Recognition of Physical Activities. In *UBICOMM 2017: The Eleventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2017. doi: ISBN9781612085982. URL https://discovery.ucl.ac.uk/id/eprint/10039438/.

[43] Alireza Fathi, Yin Li, and James M. Rehg. Learning to Recognize Daily Actions Using Gaze. In *Proceedings of the 12th European conference on Computer Vision - Volume Part I*, pages 314–327. Springer-Verlag, 2012. ISBN 978-3-642-33717-8. doi: 10.1007/978-3-642-33718-5{_}23. URL http://link.springer.com/10.1007/978-3-642-33718-5_23.

[44] Andreas Bulling, Jamie A. Wardz, Hans Gellersenz, and Gerhard Tröstery. Eye movement analysis for activity recognition. In *ACM International Conference*

Proceeding Series, pages 41–50, 2009. ISBN 9781605584317. doi: 10.1145/1620545.1620552.

[45] Kathleen A Turano, Duane R Geruschat, and Frank H Baker. Oculomotor strategies for the direction of gaze tested with a real-world activity. *Vision Research*, 43(3):333–346, 2 2003. ISSN 00426989. doi: 10.1016/S0042-6989(02)00498-4. URL http://www.ncbi.nlm.nih.gov/pubmed/12535991.

[46] Brian T. Sullivan, Leif Johnson, Constantin A. Rothkopf, Dana Ballard, and Mary Hayhoe. The role of uncertainty and reward on eye movements in a virtual driving task. *Journal of Vision*, 12(13), 2012. ISSN 15347362. doi: 10.1167/12.13.19.

[47] Leif Johnson, Brian Sullivan, Mary Hayhoe, and Dana Ballard. Predicting human visuomotor behaviour in a driving task. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1636), 2 2014. ISSN 09628436. doi: 10.1098/rstb.2013.0044.

[48] Constantin A. Rothkopf, Dana H. Ballard, and Mary M. Hayhoe. Task and context determine where you look. *Journal of Vision*, 7(14), 12 2007. ISSN 15347362. doi: 10.1167/7.14.16.

[49] David Noton and Lawrence Stark. Eye Movements and Visual Perception. *Scientific American*, 224(6):34–43, 1971. ISSN 00368733, 19467087. URL http://www.jstor.org/stable/24922750.

[50] Thomas C. Kübler, Enkelejda Kasneci, and Wolfgang Rosenstiel. SubsMatch: Scanpath similarity in dynamic scenes based on subsequence frequencies. In *Eye Tracking Research and Applications Symposium (ETRA)*, pages 319–322. Association for Computing Machinery, 2014. ISBN 9781450327510. doi: 10.1145/2578153.2578206.

[51] Thomas C. Kubler, Dennis R. Bukenberger, Judith Ungewiss, Alexandra Worner, Colleen Rothe, Ulrich Schiefer, Wolfgang Rosenstiel, and Enkelejda Kasneci. Towards automated comparison of eye-tracking recordings in dynamic scenes. In *EUVIP 2014 - 5th European Workshop on Visual Information Processing*. Institute of Electrical and Electronics Engineers Inc., 1 2015. ISBN 9781479945726. doi: 10.1109/EUVIP.2014.7018371.

[52] Yu Chen and D. H. Ballard. Learning to recognize human action sequences. In *Proceedings - 2nd International Conference on Development and Learning, ICDL 2002*, 2002.

[53] Rowel Atienza and Alexander Zelinsky. Intuitive human-robot interaction through active 3D gaze tracking. *Springer Tracts in Advanced Robotics*, 15:172–181, 2005. ISSN 16107438. doi: 10.1007/11008941{_}19. URL http://link.springer.com/10.1007/11008941_19.

[54] Katherine M. Tsui, Aman Behal, David Kontak, and Holly A. Yanco. I want that: Human-in-the-loop control of a wheelchair-mounted robotic arm. *Applied Bionics and Biomechanics*, 8(1):127–147, 2011. ISSN 17542103. doi: 10.3233/ABB-2011-0004.

[55] Songpo Li, Xiaoli Zhang, and Jeremy D. Webb. 3-D-Gaze-Based Robotic Grasping Through Mimicking Human Visuomotor Function for People with Motion Impairments. *IEEE Transactions on Biomedical Engineering*, 64(12): 2824–2835, 12 2017. ISSN 15582531. doi: 10.1109/TBME.2017.2677902. URL https://ieeexplore.ieee.org/document/7870669/.

[56] Pavel Orlov, Ali Shafti, Chaiyawan Auepanwiriyakul, Noyan Songur, and A. Aldo Faisal. A Gaze-contingent Intention Decoding Engine for human augmentation. In *Eye Tracking Research and Applications Symposium (ETRA)*, pages 1–3, New York, New York, USA, 2018. ACM Press. ISBN 9781450357067. doi: 10.1145/3204493.3208350. URL http://dl.acm.org/citation.cfm?doid=3204493.3208350.

[57] Ming Yao Wang, Alexandros A. Kogkas, Ara Darzi, and George P. Mylonas. Free-View, 3D Gaze-Guided, Assistive Robotic System for Activities of Daily Living. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2355–2361. Institute of Electrical and Electronics Engineers Inc., 12 2018. ISBN 9781538680940. doi: 10.1109/IROS.2018.8594045.

[58] Lei Shi, Cosmin Copot, and Steve Vanlanduit. Application of Visual Servoing and Eye Tracking Glass in Human Robot Interaction: A case study. In *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, pages 515–520. IEEE, 10 2019. ISBN 978-1-7281-0699-1. doi: 10.1109/ICSTCC.2019.8886064. URL https://ieeexplore.ieee.org/document/8886064/.

[59] Mohamad A. Eid, Nikolas Giakoumidis, and Abdulmotaleb El Saddik. A Novel Eye-Gaze-Controlled Wheelchair System for Navigating Unknown Environments: Case Study with a Person with ALS. *IEEE Access*, 4:558–573, 2016. ISSN 21693536. doi: 10.1109/ACCESS.2016.2520093. URL http://ieeexplore.ieee.org/document/7394111/.

[60] Ikuhisa Mitsugami, Norimichi Ukita, and Masatsugu Kidode. Robot navigation by eye pointing. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3711 LNCS, pages 256–267, 2005. ISBN 3540290346. doi: 10.1007/11558651{_}26.

[61] Irene Tong, Omid Mohareri, Samuel Tatasurya, Craig Hennessey, and Septimiu Salcudean. A retrofit eye gaze tracker for the da Vinci and its integration in task execution using the da Vinci Research Kit. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2015-Decem, pages 2043–2050. IEEE, 9 2015. ISBN 9781479999941. doi: 10.1109/IROS.2015.7353648.

URL http://ieeexplore.ieee.org/document/7353648/.

[62] David P. McMullen, Guy Hotson, Kapil D. Katyal, Brock A. Wester, Matthew S. Fifer, Timothy G. McGee, Andrew Harris, Matthew S. Johannes, R. Jacob Vogelstein, Alan D. Ravitz, William S. Anderson, Nitish V. Thakor, and Nathan E. Crone. Demonstration of a semi-autonomous hybrid brain-machine interface using human intracranial EEG, eye tracking, and computer vision to control a robotic upper limb prosthetic. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):784–796, 2014. ISSN 15344320. doi: 10.1109/TNSRE.2013.2294685.

[63] Henny Admoni and Siddhartha Srinivasa. Predicting user intent through eye gaze for shared autonomy. In *AAAI Fall Symposium - Technical Report*, volume FS-16-01 -, pages 298–303, 2016. ISBN 9781577357759. URL http://hennyadmoni.com/documents/admoni2016aaaifs.pdf.

[64] Stefanos Nikolaidis, Enkelejda Kasneci, and Siddhartha Srinivasa. Leveraging Eye Tracking and Physiological Signals for Fluent Human Robot Collaboration. *CARIS workshop at IROS*, 2017.

[65] Janis Stolzenwald and Walterio W Mayol-Cuevas. Rebellion and Obedience: The Effects of Intention Prediction in Cooperative Handheld Robots. 2019. URL https://arxiv.org/abs/1903.08158.

[66] Reem K. Al-Halimi and Medhat Moussa. Performing Complex Tasks by Users with Upper-Extremity Disabilities Using a 6-DOF Robotic Arm: A Study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(6):686–693, 6 2017. ISSN 15344320. doi: 10.1109/TNSRE.2016.2603472.

[67] Kinova Robotics, Inc. Robot arms, 2020. URL http://www.kinovarobotics.com/assistive-robotics/products/robot-arms/.

[68] Pupil Labs, Inc. Pupil labs - pupil, 2017. URL https://pupil-labs.com/pupil/.

[69] Benjamin A. Newman, Reuben M. Aronson, Siddhartha S. Srinivasa, Kris Kitani, and Henny Admoni. HARMONIC: A Multimodal Dataset of Assistive Human-Robot Collaboration. *ArXiv e-prints*, July 2018.

[70] Michael Young, Christopher Miller, Youyi Bi, Wei Chen, and Brenna D Argall. Formalized Task Characterization for Human-Robot Autonomy Allocation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6044–6050, 2019. doi: 10.1109/icra.2019.8793475. URL https://cpb-us-e1.wpmucdn.com/sites.northwestern.edu/dist/5/1812/files/2019/06/19icra_young.pdf.

[71] Cheng Shiu Chung, Hongwu Wang, and Rory A. Cooper. Autonomous function of wheelchair-mounted robotic manipulators to perform daily activities. In *IEEE*

International Conference on Rehabilitation Robotics, 2013. ISBN 9781467360241. doi: 10.1109/ICORR.2013.6650378.

[72] Michael F. Land and Mary Hayhoe. In what ways do eye movements contribute to everyday activities? *Vision Research*, 41(25):3559–3565, 2001.

[73] Barbara Sivak and Christine L. MacKenzie. Integration of visual information and motor output in reaching and grasping: The contributions of peripheral and central vision. *Neuropsychologia*, 1990. ISSN 00283932. doi: 10.1016/0028-3932(90)90143-C.

[74] iMotions, Inc. Tobii pro eye tracking glasses 2 - imotions, 2019. URL https://imotions.com/hardware/tobii-eye-tracking-glasses-2/.

[75] Carlos Elmadjian, Pushkar Shukla, Antonio Diaz Tula, and Carlos H. Morimoto. 3D gaze estimation in the scene volume with a head-mounted eye tracker. In *Proceedings - COGAIN 2018: Communication by Gaze Interaction*, pages 1–9, New York, New York, USA, 2018. ACM Press. ISBN 9781450357906. doi: 10.1145/3206343.3206351. URL http://dl.acm.org/citation.cfm?doid=3206343.3206351.

[76] Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. CalibMe: Fast and unsupervised eye tracker calibration for gaze-based pervasive human-computer interaction. In *Conference on Human Factors in Computing Systems - Proceedings*, volume 2017-May, pages 2594–2605. Association for Computing Machinery, 5 2017. ISBN 9781450346559. doi: 10.1145/3025453.3025950.

[77] Dario D Salvucci and Joseph H Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the symposium on Eye tracking research & applications - ETRA '00*, pages 71–78, 2000. ISBN 1581132808. doi: 10.1145/355017.355028. URL https://dl.acm.org/doi/10.1145/355017.355028.

[78] Enkelejda Kasneci, Gjergji Kasneci, Thomas C. Kübler, and Wolfgang Rosenstiel. The applicability of probabilistic methods to the online recognition of fixations and saccades in dynamic scenes. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14*, pages 323–326, New York, New York, USA, 2014. ACM Press. ISBN 9781450327510. doi: 10.1145/2578153.2578213. URL http://dl.acm.org/citation.cfm?doid=2578153.2578213.

[79] Thiago Santini, Wolfgang Fuhl, Thomas Kübler, and Enkelejda Kasneci. Bayesian identification of fixations, saccades, and smooth pursuits. In *Eye Tracking Research and Applications Symposium (ETRA)*, volume 14, pages 163–170. Association for Computing Machinery, 3 2016. ISBN 9781450341257. doi: 10.1145/2857491.2857512.

[80] Monica S. Castelhano, Michael L. Mack, and John M. Henderson. Viewing task influences eye movement control during active scene perception. *Journal of*

Vision, 9(3), 3 2009. ISSN 15347362. doi: 10.1167/9.3.6.

[81] Thies Pfeiffer and Patrick Renner. EyeSee3D: a low-cost approach for analyzing mobile 3D eye tracking data using computer vision and augmented reality technology. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14*, pages 369–376, New York, New York, USA, 2014. ACM Press. ISBN 9781450327510. doi: 10.1145/2578153.2628814. URL http://dl.acm.org/citation.cfm?doid=2578153.2628814.

[82] Thies Pfeiffer, Patrick Renner, and Nadine Pfeiffer-Leßmann. EyeSee3D 2.0. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*, pages 189–196, New York, New York, USA, 2016. ACM Press. ISBN 9781450341257. doi: 10.1145/2857491.2857532. URL http://dl.acm.org/citation.cfm?doid=2857491.2857532.

[83] Kakeru Hagihara, Keiichiro Taniguchi, Irshad Abibouraguimane, Yuta Itoh, Keita Higuchi, Jiu Otsuka, Maki Sugimoto, and Yoichi Sato. Object-wise 3d gaze mapping in physical workspace. In *Proceedings of the 9th Augmented Human International Conference*, AH '18, pages 25:1–25:5, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5415-8. doi: 10.1145/3174910.3174921. URL http://doi.acm.org/10.1145/3174910.3174921.

[84] S. Li, X. Zhang, and J. D. Webb. 3-d-gaze-based robotic grasping through mimicking human visuomotor function for people with motion impairments. *IEEE Transactions on Biomedical Engineering*, 64(12):2824–2835, Dec 2017. ISSN 0018-9294. doi: 10.1109/TBME.2017.2677902.

[85] Lucas Paletta, Katrin Santner, Gerald Fritz, Albert Hofmann, Gerald Lodron, Georg Thallinger, and Heinz Mayer. Facts - a computer vision system for 3d recovery and semantic mapping of human factors. In Mei Chen, Bastian Leibe, and Bernd Neumann, editors, *Computer Vision Systems*, pages 62–72, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[86] Karishma Singh, Mahmoud Kalash, and Neil Bruce. Capturing real-world gaze behaviour: Live and unplugged. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pages 20:1–20:9, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5706-7. doi: 10.1145/3204493.3204528. URL http://doi.acm.org/10.1145/3204493.3204528.

[87] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Human behavior modeling with maximum entropy inverse optimal control. In *AAAI Spring Symposium - Technical Report*, volume SS-09-04, pages 92–97, 2009. ISBN 9781577354116. URL http://www.cs.cmu.edu/~bziebart/publications/human-behavior-bziebart.pdf.

[88] Anca D. Dragan, Kenton C.T. Lee, and Siddhartha S. Srinivasa. Legibility

and predictability of robot motion. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 301–308, 2013. ISBN 9781467330558. doi: 10.1109/HRI.2013.6483603.

[89] Thomas C Kübler, Shahram Eivazi, and Enkelejda Kasneci. Automated Visual Scanpath Analysis Reveals the Expertise Level of Micro-neurosurgeons. Technical report.