UNIVERSITY OF CALIFORNIA

Los Angeles

Eye Gaze-based Approaches to Recognize Human Intent
for Shared Autonomy Control of Robot Manipulators

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Xiaoyu Wang

2021

ABSTRACT OF THE DISSERTATION

Eye Gaze-based Approaches to Recognize Human Intent
for Shared Autonomy Control of Robot Manipulators

by

Xiaoyu Wang
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2021
Professor Veronica Santos, Chair

Robots capable of robust, real-time recognition of human intent during manipulation tasks could be used to enhance human-robot collaboration for innumerable applications. Eye gaze-based control interfaces offer a non-invasive way to infer intent and reduce the cognitive burden on operators of complex robots. Eye gaze is traditionally used for “gaze triggering” (GT) in which staring at an object, or sequence of objects, triggers pre-programmed robotic movements. Our long-term objective is to leverage eye gaze as an intuitive way to infer human intent, advance action recognition for shared autonomy control, and enable seamless human-robot collaboration not yet possible with state-of-the-art gaze-based methods.

In Study #1, we identified features from 3D gaze behavior for use by machine learning classifiers of action recognition. We investigated gaze behavior and gaze-object interactions as participants performed the bimanual activity of preparing a powdered drink. We generated 3D gaze saliency maps and used characteristic gaze object sequences to demonstrate an action recognition algorithm.

In Study #2, we introduced a classifier for recognizing action primitives, which we defined as triplets having a verb, “target object,” and “hand object.” Using novel 3D gaze-related features, a recurrent neural network was trained to recognize a verb and target object. The gaze object angle and its rate of change enabled accurate recognition and a reduction in the observational latency of the classifier. Using a non-specific approach to indexing objects, we demonstrated modest generalizability of the classifier across activities.

In Study #3, we introduced a neural network-based “action prediction” (AP) mode into a shared autonomy framework capable of 3D gaze reconstruction, real-time intent recognition, object localization, obstacle avoidance, and dynamic trajectory planning. Upon extracting gaze-related features, the AP model recognized, and often predicted, the operator’s intended action primitives. The AP control mode, often preferred over a state-of-the-art GT mode, enabled more seamless human-robot collaboration.

In summary, we developed machine learning-based action recognition methods using novel 3D gaze-related features to enhance the shared autonomy control of robot manipulators. Our methods can serve as a foundation for further enhancement with complementary sensory feedback such as computer vision and tactile sensing.

The dissertation of Xiaoyu Wang is approved.

Song-Chun Zhu

Jacob Rosen

Tetsuya Iwasaki

Veronica Santos, Committee Chair

University of California, Los Angeles

2021

To mom and dad . . .

who through their hard work, perseverance, and endless support

have always embodied the role-model I want to become

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, it is my pleasure to express my sincere gratitude to my advisor, Dr. Veronica J. Santos. I feel extremely fortunate to work under her supervision. Her relentless support, invaluable advice, and continuous encouragement have helped me, to a great extent, to overcome all kinds of difficulties of my academic research and daily life. This work would not have been possible without her support. I have benefit a lot from her immense knowledge and intellectual guidance.

I would like thank my committee members Dr. Song-chun Zhu, Dr. Tetsuya Iwasaki, and Dr. Jacob Rosen for providing me with patient advice and guidance through the research process, especially during my proposal and final defense. I am grateful for the opportunity to learn from each of them, no matter it is in classes, seminars, or in research. I hope I can take the support they provided to me to pay it forward whenever I can.

For Chapter 2, I would like thank Allison Walters for her contributions to the initial experimental protocol and equipment setup. Thanks to Yuzhen Chen for discussion concerning 3D reconstruction of the gaze vector. For Chapter 3, thanks to Dr. Eunsuk Chong, Yi Zheng, and Eric Peltola for brainstorming ideas of intent inference model using machine learning methods. For Chapter 4, thanks to Muhan Hou for his contributions to the initial experimental setup, Alireza haji Fathaliyan for his contributions to the initial experimental design and data collection, and Yi Zheng for discussions on early drafts of this manuscript.

The material presented in Chapters 2, 3, and 4 were supported by National Science Foundation Award 1461547, 1463960, and Office of Naval Research Award (ONR) N00014-16-1-2468. Any opinions, findings, conclusions, or recommendations are those of the authors and do not necessarily represent the official views, opinions, or policies of the funding agencies.

A special thanks to all the members of the UCLA Biomechatronics Laboratory. They have created a research environment of mutual help and friendship. They have taught and inspired me a lot, no matter it is in identifying a scientific problem, designing and constructing experimental setup, solving engineering problems, or presenting academic achievements with enthusiasm. A special thanks to Alireza Haji Fathaliyan. We encouraged each other and cooperated closely in brainstorming research ideas and overcoming technical issues. Thanks to Yi Zheng. He has always been a good company when I debug the robot in the lab. Thanks to Dr. Kenneth Gutierrez. He provided me with great help in writing the dissertation. Thanks to Alexis Block. She gave me many valuable suggestions in preparing my defense slides.

A special thanks to my parents for their endless support. They have always stood behind me, and this was no exception.

VITA

2016–Present Graduate Student Researcher, UCLA Biomechatronics Lab, Mechanical and Aerospace Engineering Department, UCLA, Los Angeles, California

2016, '18, '19 Teaching Assistant, Physics Department and Mechanical and Aerospace Engineering Department, UCLA, Los Angeles, California

2016 M.S. Aerospace Engineering, UCLA, Los Angeles, California

2014 B.S. Control Science and Engineering, Harbin Institute of Technology, Harbin, China

2012 Exchange student, Engineering School, National University of Singapore, Singapore

PUBLICATIONS

Wang, X., Fathaliyan, A., and Santos, V.J. ”Toward shared autonomy control schemes for human-robot systems: Action primitive recognition using eye gaze features.” in Frontiers in Neurorobotics, vol. 14, Article 567571, Oct 2020.

Fathaliyan, A., **Wang, X.**, and Santos, V.J. ”Exploiting 3D gaze tracking for action recognition during bimanual manipulation to enhance human-robot collaboration.” in Frontiers in Robotics and AI, vol. 5, Article 25, Apr 2018.

Fathaliyan, A., **Wang, X.**, Bazargan, S., and Santos, V.J. ”Hand-object kinematics and gaze fixation during bimanual tasks.” in Proceedings of the Annual Meeting of the American Society of Biomechanics, Boulder, CO, August 9, 2017.

CHAPTER 1

Introduction

1.1 Motivation

The eyes are the window to the mind. Eye gaze contains rich information about mental state, and it can reflect an individual's thoughts [1, 2]. Eye tracking technology can be used to quantify natural gaze behaviors and extract information about gaze fixation, for example. Eye tracking also provides a non-verbal and non-invasive interface for people to communicate their intent to a computer or robot. Eye tracking has been applied in a broad range of fields such as marketing, skill training/assessment, cognitive psychology, and robotics.

1.1.1 Applications of eye tracking

In the field of marketing and advertising, gaze fixation distribution on product packages and webpages has been tracked for customer data analysis. Zamani and Amin have verified the strong relationships between gaze fixation count, gaze fixation duration, and customers' purchase decisions [3]. Retailers could leverage the eye tracking technique to optimize product packaging and retail shelf design to better capture and direct consumers' attention to cues that ultimately lead to purchases [4]. Website design agencies could better understand mainstream users' browsing habits through eye tracking data and display important information in positions with higher visual saliency [5].

Eye tracking has been used for skill training and assessment. For instance, Khan et al. demonstrated a significant difference in gaze patterns between novice and expert surgeons as they observed a laparoscopic operation video [6]. Researchers found that expert surgeons' eyes were more focused on key target areas of the operative field, while novice surgeons' eyes often wandered from the key areas. Potentially, novice surgeons could expedite their learning curves by following the gaze patterns of expert surgeons.

The field of cognitive psychology has studied pupil dilation as a reflection of intensity-related aspects of cognitive processing [1]. Pupil dilation has been monitored in tasks such as arithmetic operations, reading comprehension [7], digit sorting, and digit span (short-term memory) [8]. In these tasks, larger pupil dilation has been reported in higher difficulty-level conditions than lower difficulty-level conditions. For instance, Bradley et al. compared pupil diameters as subjects were instructed to view pleasant, unpleasant, and neutral pictures [9]. Researchers found that pupil dilation was larger when viewing emotionally arousing pictures (no matter pleasant or unpleasant) and concluded that pupil dilation can reflect emotional arousal in addition to cognitive burden.

Considering robotics applications, eye tracking has been used as an interface for teleoperating robotic agents such as drones [10, 11, 12], wheelchairs [13], and medical robots [14]. Yu et al. designed a set of "gaze gestures" and assigned each gesture to the direct control of one degree of freedom of a drone, including speed, rotation, translation, and altitude. Raymond et al. collected gaze signals as operators used a joystick to control the movement of a wheelchair [13]. A fitting function was trained to predict joystick signals using gaze point positions as the input. With the fitting function, an operator could control the linear and angular velocity of the wheelchair using gaze commands. Li et al. developed an attention-aware robotic laparoscopic system that could recognize an operator's visual target through eye tracking signals and then automatically steer the laparoscope to focus on the target [14]. Such a gaze-based targeting system has the potential to make the execution of surgeries smoother and more efficient.

1.1.2 Gaze-based shared autonomy for human-robot collaboration

Nowadays, with the rapid advancement of robotics, computational power, and machine learning techniques, the integration of the eye tracking with robot control schemes has attracted widespread attention. As in the literature, we distinguish between direct teleoperation and shared autonomy [15]. With gaze-based teleoperation, an operator's gaze signals are mapped directly to robot positions or velocities, putting the cognitive burden on the operator [15]. In contrast, in a gaze-based shared autonomy system, an operator's gaze input is combined with semi-autonomous robot decisions in order to achieve shared goals. Specifically, a shared autonomy system could model and predict an operator's intended actions through natural eye movements and leverage robot autonomy to execute the recognized actions. Shared autonomy can effectively decrease operators' cognitive burden and make the control process more natural, intuitive, and seamless.

A robust, real-time intent recognition or prediction model is the key to an effective gaze-based shared autonomy system and should, at a minimum, satisfy the following requirements: (i) Ideally, the model should predict intent prior to the initiation of an action and not rely solely on the visual consequences of actions. (ii) In the case of assistive robotic manipulators, the intent prediction model should enable more tasks than pick-and-place alone, as activities of daily living (ADLs) require a rich set of functional behaviors.

Numerous computer vision-based studies have leveraged egocentric videos taken by head-mounted cameras or eyetrackers to recognize actions during everyday tasks [16, 17, 18, 19, 20, 21, 22, 23, 24]. However, in these studies, actions could not be successfully recognized until key visual features related to hand motions and object states (e.g. whether a lid is on a cup) were observable and available to the classification algorithm. While such classifiers are useful, they are incapable of predicting intent due to their reliance on the visual consequences of actions.

Some gaze-based shared autonomy studies have focused on the pick-and-place tasks [25, 26, 27]. In these studies, the gaze point was the only gaze-related feature utilized to predict intent. Classification methods based on support vector machines (SVMs) and partially observable Markov decision processes (POMDPs) were used to estimate a target object for pick-up or a target position for setting down a grasped object. Pick-and-place capabilities are immensely useful, but the intent prediction framework in the literature does not lend itself to expansion for other functional behaviors common to ADLs, such as pouring and stirring.

While algorithms have been developed to recognize human intent by other research groups, we were unable to find follow-on publications in which the proposed algorithms were demonstrated experimentally on real robots [25, 28, 29]. In addition, previous studies on gaze-based action recognition and shared autonomy relied only on gaze positions from two-dimensional videos captured by ego-centric cameras or eyetrackers. It is likely that some intent-relevant information could be encoded by three-dimensional (3D) spatiotemporal relationships between gaze vectors and the environment. Thus, there is a need to extract novel features from 3D gaze behaviors, develop new gaze-based intent estimation algorithms, and assess their performance in shared autonomy systems on real robots.

1.2 Contributions

The work in this dissertation presents methods for extracting novel 3D gaze-related features, designing classifiers for action recognition, and a shared autonomy control framework for collaborative human-robot systems. Moeslund et al. described human behaviors as a composition of three hierarchical levels: (i) activities, (ii) actions, and (iii) action primitives. In this dissertation, we rephrase the hierarchical levels as (i) activities, (ii) *subtasks*, and (iii) action primitives, and include subtasks and action primitives under the umbrella of actions to be recognized.

We identified gaze-related features that are extremely useful for action recognition including gaze object, gaze object sequence, gaze object angle, and gaze object angular speed. Through dynamic time warping, we were able to recognize actions at the subtask level, and through recurrent neural networks, we were able to recognize actions at the action primitive level. We successfully demonstrated the feasibility and advantages of using gaze-based action recognition algorithms to enhance the operator experience in shared autonomy systems.

Chapter 2 presents a novel method to construct a gaze vector and gaze saliency maps in 3D space, which enables the analysis of gaze behaviors and gaze-object interactions from a variety of 3D perspectives. Using dynamic time warping, we created a population-based set of characteristic gaze object sequences and demonstrated action recognition at the subtask level.

Chapter 3 presents a gaze-based action primitive recognition algorithm that can be used in shared autonomy systems that assist with activities of daily living. We define an action primitive as a triplet comprised of a verb, “target object,” and “hand object.” The algorithm leverages a long short-term memory recurrent neural network to recognize participants’ intended verb and target object. We demonstrated that the use of novel gaze-related features, such as gaze object angle and gaze object angular speed, are especially useful for accurately recognizing action primitives and reducing the observational latency of the classifier.

Chapter 4 presents a novel gaze-based shared autonomy framework for human-robot collaboration to assist with activities of daily living. The shared autonomy framework is capable of 3D gaze reconstruction, real-time intent recognition, object localization, obstacle avoidance, and dynamic trajectory planning. We describe the development and implementation of the gaze-based shared autonomy scheme on a real robot. We implemented the action primitive recognition algorithm trained in Chapter 3, which enabled the robot to recognize, and often predict, the operator's intended actions. We show that the "action prediction" mode, often preferred over a state-of-the-art "gaze trigger" mode, enabled more seamless human-robot collaboration

Chapter 5 summarizes this dissertation and presents potential opportunities for enhancement as future work.

CHAPTER 2

Exploiting 3D Gaze Tracking for Action Recognition During Bimanual Manipulation to Enhance Human-robot Collaboration

This chapter was based on work published in the journal Frontiers in Robotics and AI [30].

2.1 Abstract

Human-robot collaboration could be advanced by facilitating the intuitive, gaze-based control of robots, and enabling robots to recognize human actions, infer human intent, and plan actions that support human goals. Traditionally, gaze tracking approaches to action recognition have relied upon computer vision-based analyses of 2D egocentric camera videos. The objective of this study was to identify useful features that can be extracted from 3D gaze behavior and used as inputs to machine learning algorithms for human action recognition. We investigated human gaze behavior and gaze-object interactions in 3D during the performance of a bimanual, instrumental activity of daily living: the preparation of a powdered drink. A marker-based motion capture system and binocular eye tracker were used to reconstruct 3D gaze vectors and their intersection with 3D point clouds of objects being manipulated. Statistical analyses of gaze fixation duration and saccade size suggested that some actions (pouring, stirring) may require more visual attention than other actions (reach, pick up, set down, move). Three-dimensional gaze saliency maps, generated with high spatial resolution for six subtasks, appeared to encode action-relevant information. The ”gaze object sequence” was used to capture information about the identity of objects in concert with the temporal sequence in which the objects were visually regarded. Dynamic time warping barycentric averaging was used to create a population-based set of characteristic gaze object sequences that accounted for intra- and inter-subject variability. The gaze object sequence was used to demonstrate the feasibility of a simple action recognition algorithm that utilized a dynamic time warping Euclidean distance metric. Recognition accuracy results of 91.5%, averaged over the six subtasks, suggest that the gaze object sequence is a promising feature for action recognition whose impact could be enhanced through the use of sophisticated machine learning classifiers and algorithmic improvements for real-time implementation. Robots capable of robust, real-time recognition of human actions during manipulation tasks could be used to improve quality of life in the home as well as quality of work in industrial environments.

2.2 Introduction

Recognition of human motion has the potential to greatly impact a number of fields, including assistive robotics, human-robot interaction, and autonomous monitoring systems. In the home, recognition of instrumental activities of daily living (iADLs) could enable an assistive robot to infer human intent and collaborate more seamlessly with humans while also reducing the cognitive burden on the user. A wheelchair-mounted robot with such capabilities could enhance the functional independence of wheelchair users with upper limb impairments [31]. During bimanual iADLs, humans rely heavily on vision to proactively gather task-relevant visual information for planning [32]. For example, task-relevant information for manipulation could include the three-dimensional (3D) location of an object as well as its structure-related and substance-related properties, such as shape and weight, respectively [33]. Saccades typically precede body movement [34] and reflect one's stratey for successful completion of a task.

The relationships between human vision, planning, and intent have inspired roboticists to adopt similar vision-based principles for planning robot movements and to use human gaze tracking for the intuitive control of robot systems. For instance, gaze fixation data collected during the human navigation of rocky terrain have been used to inspire the control of bipedal robots, specifically for the identification and selection of foot placement locations during traversal of rough terrain [35]. Human eye tracking data have also been used in the closed loop control of robotic arms. Recently, [36] demonstrated how 3D gaze tracking could be used to enable individuals with impaired mobility to control a robotic arm in an intuitive manner. Diverging from traditional gaze tracking approaches that leverage two-dimensional (2D) egocentric camera videos, Li et al. presented methods for estimating object location and pose from gaze points reconstructed in 3D. A visuomotor grasping model was trained on gaze locations in 3D along with grasp configurations demonstrated by unimpaired subjects. The model was then used for robot grasp planning driven by human 3D gaze.

In this work, we consider how human eye movements and gaze behavior may encode intent and could be used to inform or control a robotic system for the performance of bimanual tasks. Unlike repetitive, whole-body motions such as walking and running, iADLs can be challenging for autonomous recognition systems for multiple reasons. For instance, human motion associated with iADLs is not always repetitive, often occurs in an unstructured environment, and can be subject to numerous visual occlusions by objects being manipulated as well as parts of the human body. Prior studies on recognition of iADLs often applied computer vision-based approaches to images and videos captured via egocentric cameras worn by human subjects. Video preprocessing methods typically consist of first subtracting the foreground and then detecting human hands, regions of visual interest, and objects being manipulated [17, 19, 37, 38].

A variety of methods have been presented for feature extraction for use in machine learning classifiers. In some studies, hand-hand, hand-object, and/or object-object relationships have been leveraged [16, 18, 21]. The state of an object (e.g., open vs. closed) has been used as a feature of interest [20].Another study leveraged a saliency-based method to estimate gaze position, identify the "gaze object" (the object of visual regard), and recognize an action [22].Other studies have employed eye trackers in addition to egocentric cameras; researchers have reported significant improvements in action recognition accuracy as a result of the additional gaze point information [16, 19].

In the literature, the phrase "saliency map" has been used to reference a topographically arranged map that represents visual saliency of a corresponding visual scene [39]. In this work, we will refer to "gaze saliency maps" as heat maps that represent gaze fixation behaviors. 2D gaze saliency maps have been effectively employed for the study of gaze behavior while viewing and mimicking the grasp of objects on a computer screen [40]. Belardinelli et al. showed that gaze fixations are distributed across objects during action planning and can be used to anticipate a user's intent with the object (e.g., opening vs. lifting a teapot). While images of real world objects were presented, subjects were only instructed to mimic actions. In addition, since such 2D gaze saliency maps were constructed from a specific camera perspective, they cannot be easily generalized to other views of the same object. One of the objectives of this work was to construct gaze saliency maps in 3D that could enable gaze behavior analyses from a variety of perspectives. Such 3D gaze saliency maps could be mapped to 3D point clouds trivially obtained using low-cost RGB-D computer vision hardware, as is common in robotics applications. Furthermore, given that all manipulation tasks occur in three dimensions, 3D gaze saliency maps could enable additional insights into action-driven gaze behaviors. Although our experiments were conducted in an artificial lab setting using an uncluttered object scene, the experiment enabled subjects to perform actual physical manipulations of the object as opposed to only imagining or mimicking the manipulations, as in [40].

The primary objective of this study was to extract and rigorously evaluate a variety of 3D gaze behavior features that could be used for human action recognition to benefit human–robot collaborations. Despite the increasing use of deep learning techniques for end-to-end learning and autonomous feature selection, in this work, we have elected to consider the potential value of independent features that could be used to design action recognition algorithms in the future. In this way, we can consider the physical meaning, computational expense, and value added on a feature-by-feature basis. In Section "Materials and Methods," we describe the experimental protocol, methods for segmenting actions, analyzing eye tracker data, and constructing 3D gaze vectors and gaze saliency maps. In Section "Results," we report trends in eye movement characteristics and define the "gaze object sequence." In Section "Discussion," we discuss observed gaze behaviors and the potential and practicalities of using gaze saliency maps and gaze object sequences for action recognition. Finally, in Section "Conclusion," we summarize our contributions and suggest future directions.

2.3 Materials and Methods

2.3.1 Experimental Protocol

This study was carried out in accordance with the recommendations of the UCLA Institutional Review Board with written informed consent from all subjects. All subjects gave written informed consent in accordance with the Declaration of Helsinki. The protocol was approved by the UCLA Institutional Review Board. A total of 11 subjects (nine males, two females; aged 18–28 years) participated in the study, whose preliminary results were first reported in [41]. According to a handedness assessment [42], two subjects were "pure right handers," seven subjects were "mixed right handers," and two subjects were "neutral."

Subjects were instructed to perform a bimanual tasks involving everyday objects and actions. In this work, we focus on one bimanual task that features numerous objects and subtasks: the preparation of a powdered drink. To investigate how the findings of this study may generalize to other iADL tasks, we plan to apply similar analyses to other bimanual tasks in the future. The objects for the drink preparation task were selected from the benchmark Yale-CMU-Berkeley (YCB) Object Set [43]: mug, spoon, pitcher, and pitcher lid. The actions associated with these objects were reach for, pick up, set down, move, stir, scoop, drop, insert, and pour.

Subjects were instructed to repeat the task four times with a 1 min break between each trial. The YCB objects were laid out and aligned on a table (adjusted to an ergonomic height for each subject) as shown in Figure 1. The experimental setup was reset prior to each new trial. Subjects were instructed to remove a pitcher lid, stir the contents of the pitcher, which contained water only (the powdered drink was imagined), and transfer the drink from the pitcher to the mug in two different ways. First, three spoonfuls of the drink were to be transferred from the pitcher to the mug using a spoon. Second, the pitcher lid was to be closed to enable to pouring of the drink from the pitcher to the mug until the mug was filled to two-third of its capacity. In order to standardize the instructions provided to subjects, the experimental procedure was demonstrated via a prerecorded video.

Subjects wore an ETL-500 binocular, infrared, head-mounted eye tracker (ISCAN, Inc., Woburn, MA, USA) that tracked their visual point of regard, with respect to a head-mounted egocentric scene camera, at a 60 Hz sampling frequency. Calibration data suggest that the accuracy and precision of the eye tracker are approximately 1.43° and 0.11°, respectively. Six T-Series cameras sampled at 100 Hz and a Basler/Vue video camera (Vicon, Culver City, CA, USA) were used to track the motion of the subjects and YCB objects (Figure 1). Retroreflective markers were attached to the YCB objects, eye tracker, and subjects' shoulders, upper arms, forearms, and hands (dorsal aspects). Visual distractions were minimized through the use of a blackout curtain that surrounded the subject's field of view.

2.3.2 Action Segmentation: Task, Subtask, and Action Unit Hierarchy

[34] reported on gaze fixation during a tea-making task. In that work, a hierarchy of four activity levels was considered: "make the tea" (level 1), "prepare the cups" (level 2), "fill the kettle" (level 3), and "remove the lid" (level 4). [44] reported on a brownie-making task and divided the task into 29 actions, such as "break one egg" and "pour oil in cup." Adopting a similar approach as these prior works, we defined an action hierarchy using a task–subtask–action unit format (Table 1). Subtasks were defined similar to Land et al.'s "4th level activities" while the action units were defined according to hand and object kinematics. All subjects performed all six subtasks listed in Table 1, but not all subjects performed all action units. For example, a couple of subjects did not reach for the pitcher

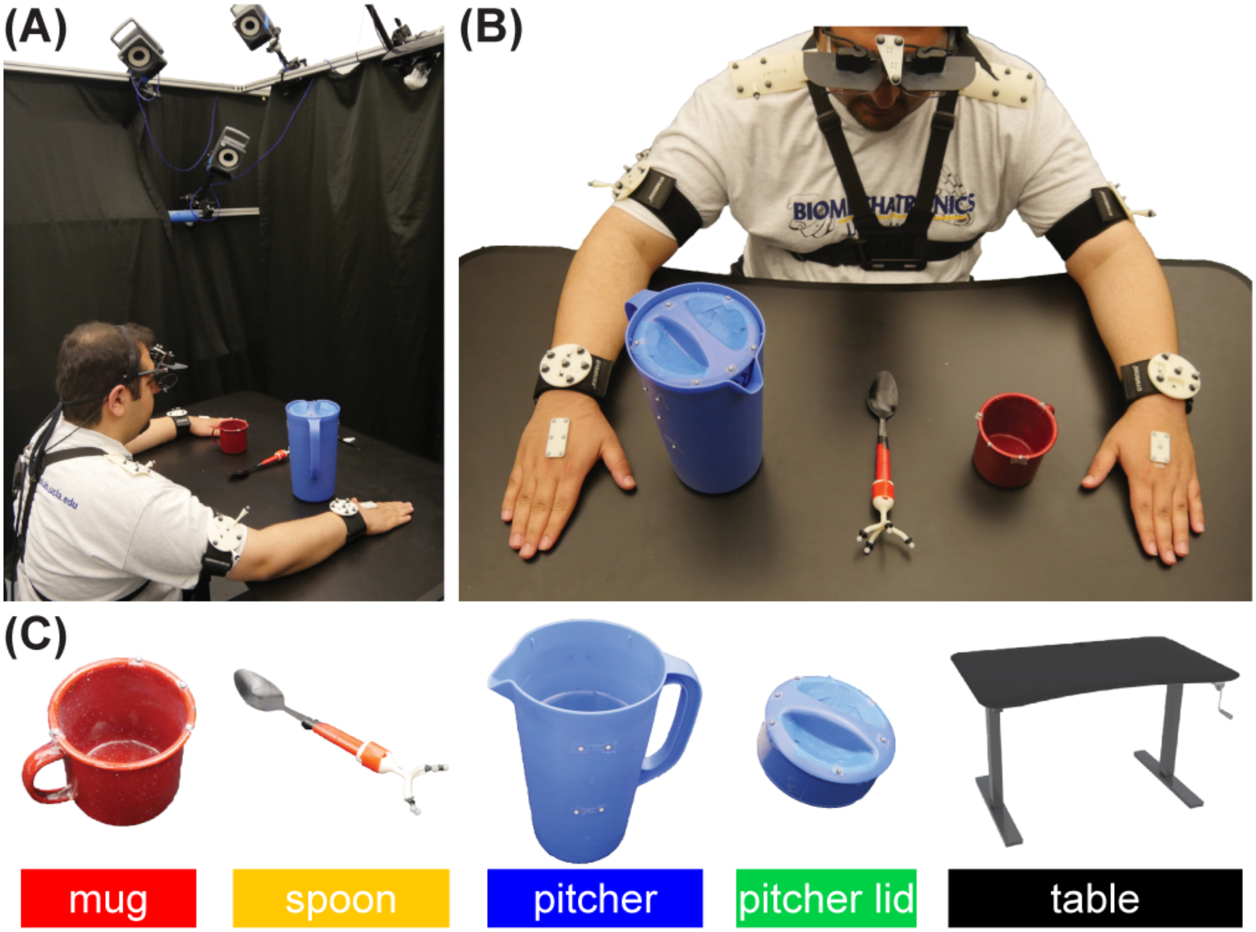


Figure 2.1: (A) Each subject was seated in the motion capture area. A blackout curtain was used to minimize visual distractions. (B) The subject wore a head-mounted eye tracker. Motion capture markers were attached to the Yale-CMU-Berkeley objects, the eye tracker, and subjects' upper limbs. Each trial used the object layout shown. (C) Retroreflective markers were placed on a mug, spoon, pitcher, pitcher lid, and table. These objects will be referenced using the indicated color code throughout this manuscript. The subject shown in panels (A,B) has approved of the publication of these images.

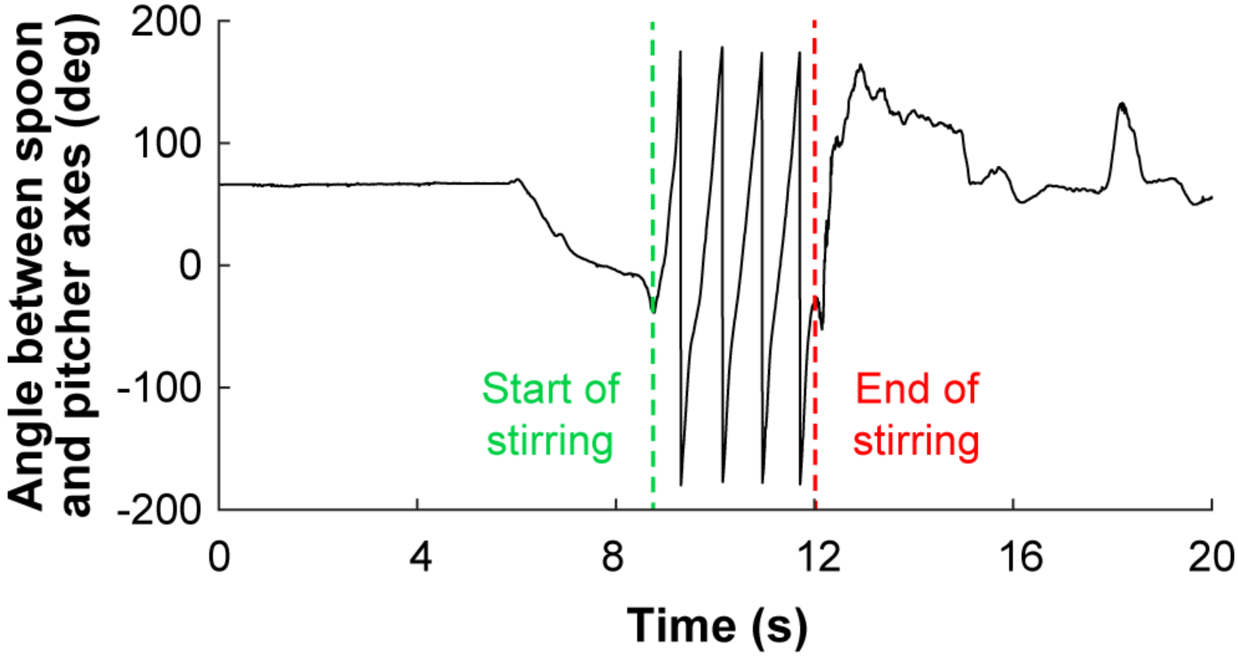


Figure 2.2: The repetitive nature of the spoon's kinematics with respect to the pitcher was used to identify the start and end of the action unit "stir inside pitcher." Although the spoon was not manipulated until approximately 6 s had elapsed in the representative trial shown, the full trial is provided for completeness.

during Subtask 2 ("move spoon into pitcher").

The start and end time of each action unit were identified according to hand and object kinematics and were verified by observing the egocentric video recorded from the eye tracker. For example, the angle of the spoon's long axis with respect to the pitcher's long axis and the repetitive pattern of the angle were used to identify the beginning and end of the action unit "stir inside pitcher" (Figure 2).

2.3.3 Gaze Fixation and Saccade Labeling

Saccadic movements of the eye were discovered by Edwin Landott in 1890 while studying eye movements during reading [45]. According to Kandel et al., saccadic eye movements are characterized by "jerky movements followed by a short pause" or "rapid movements between fixation points." In our study, saccades were detected using the angular velocity of the reconstructed gaze vector (see 3D Gaze Vector and Gaze Saliency Map Construction) and intervals between saccades that exceeded 200 ms were labeled as gaze fixations, as in [46]. As described previously, the beginning and end of action units were defined based on hand and object kinematics. A heuristic approach, as outlined in Figure 3, was used to associate gaze fixation periods and saccades in the eye tracker data with action units. A given gaze fixation period was associated with a specific action unit if the gaze fixation period overlapped with the action unit period ranging from 0.3 to 0.7 T, where T was the duration of the specific action unit. A given saccade was associated with a specific action unit if the saccade occurred during the action unit period ranging from -0.2 to 0.8 T. Saccade to action unit associations were allowed prior to the start of the action unit (defined from hand and object kinematics) based on reports in the literature that saccades typically precede related motions of the hand [34, 32]. The results of the approach presented in Figure 3 were verified through careful comparison with egocentric scene camera videos recorded by the eye tracker.

2.3.4 3D Gaze Vector and Gaze Saliency Map Construction

The eye tracker provided the 2D pixel coordinates of the gaze point with respect to the image plane of the egocentric scene camera. The MATLAB Camera Calibration Toolbox [47, 48] and a four-step calibration procedure were used to estimate the camera's intrinsic and

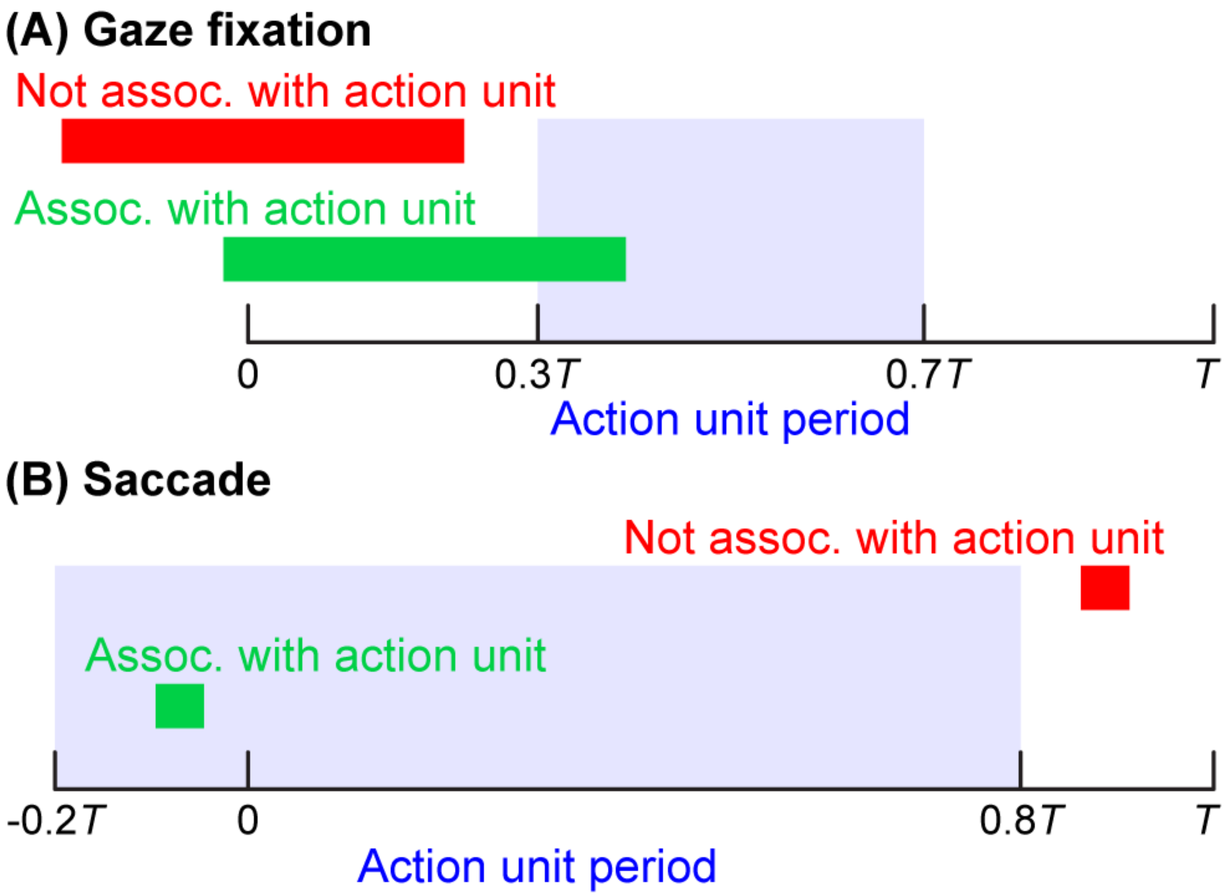


Figure 2.3: (A) A given gaze fixation period was associated with a specific action unit if the gaze fixation period overlapped with the action unit period ranging from 0.3 to 0.7 T (blue shaded region), where T was the duration of the specific action unit. (B) A given saccade was associated with a specific action unit if the saccade occurred during the action unit period ranging from -0.2 to 0.8 T.

	Subtask 1: remove pitcher lid	**Subtask 2: move spoon into pitcher**	**Subtask 3: stir inside pitcher**	**Subtask 4: transfer liquid from pitcher to mug using spoon**	**Subtask 5: replace pitcher lid**	**Subtask 6: pour liquid into mug**
Action units	Reach for pitcher lid	Reach for pitcher	Stir	Scoop inside pitcher	Reach for pitcher lid	Reach for mug
	Reach for pitcher	Reach for spoon		Reach for mug	Reach for pitcher	Pick up mug
	Pick up pitcher lid	Pick up spoon		Move mug to pitcher	Pick up pitcher lid	Move mug to pitcher
	Set down pitcher lid	Move spoon		Move spoon to mug	Move pitcher lid to pitcher	Reach for pitcher handle
				Drop liquid into mug using spoon	Insert pitcher lid into pitcher	Pick up pitcher
				Set down mug		Pour liquid
				Set down spoon		Set down pitcher

Table 2.1: Six subtasks were defined for the task of making a powdered drink; action units were defined for each subtask according to hand and object kinematics.

extrinsic parameters. These parameters enabled the calculation of the pose of the 2D image plane in the 3D global reference frame. The origin of the camera frame was located using motion capture markers attached to the eye tracker. The 3D gaze vector was reconstructed by connecting the origin of the camera frame with the gaze point's perspective projection onto the image plane.

Using the reconstructed 3D gaze vector, we created 3D gaze saliency maps by assigning RGB colors to the point clouds obtained from 3D scans of the YCB objects. The point cloud for the mug was obtained from [43]. The point clouds for the pitcher, pitcher lid, and spoon were scanned with a structured-light 3D scanner (Structure Sensor, Occipital, Inc., CA, USA) and custom turntable apparatus. This was necessary because the YCB point cloud database only provides point clouds for the pitcher lid assembly and because the proximal end of the spoon was modified for the application of motion capture markers (Figure 1C). Colors were assigned to points based on the duration of their intersection with the subject's 3D gaze vector. In order to account for eye tracker uncertainty, colors were assigned to a 5 mm-radius spherical neighborhood of points, with points at the center of the sphere (intersected by the 3D gaze vector) being most intense. Color intensity for points within the sphere decreased linearly as the distance from the center of the sphere increased. Both gaze fixation and saccades were included during RGB color assignment. For each subtask, the RGB color intensity maps were summed across subjects and then normalized to the [0, 1] range, with 0 as black and 1 as red. The normalization was performed with all task-relevant objects considered simultaneously and not on an object-specific basis. This enabled the investigation of the relative visual importance of each object for each subtask.

2.4 Results

2.4.1 Eye Movements: Gaze Fixation Duration and Saccade Size

Gaze fixation duration and saccade size have previously been identified as important features for gaze behaviors during iADLs. As in [49], we use "saccade size" to refer to the angle spanned by a single saccade. [34] reported overall trends and statistics for the entire duration of a tea-making task. However, information about dynamic changes in gaze behavior is difficult to extract and analyze when eye tracker data are convolved over a large period of time. In order to address eye movements at a finer level of detail, we investigated trends in gaze fixation duration and saccade size at the action unit level. Gaze fixation duration data were normalized by summing the durations of gaze fixation periods that belonged to the same action unit and then dividing by the total duration of that action unit. This normalization was performed to minimize the effect of action unit type, such as reaching vs. stirring, on gaze fixation duration results. Gaze fixation duration and saccade size were analyzed according to groupings based on six common action unit verbs: "reach," "pick up," "set down," "move," "pour," and "stir" (Figure 4). "Drop" and "insert" were excluded, as they occurred infrequently and their inclusion would have further reduced the power of the

Fixation Saccade	Reach	Pick up	Set down	Move	Pour	Stir
Reach		0.012	0.050	3e-6*	0.030	2e-13*
Pick up	0.707		0.450	5e-10*	0.462	3e-12*
Set down	0.242	0.496		3e-10*	0.938	2e-9*
Move	0.666	0.992	0.432		9e-8*	9e-23*
Pour	1e-10*	6e-9*	2e-8*	4e-10*		3e-8*
Stir	3e-9*	1e-7*	4e-7*	1e-8*	0.512	

Asterisks indicate the t-tests that were statistically significant for a Bonferroni-corrected $\alpha = 0.003$.

Table 2.2: The lower left triangle of the table (shaded in gray) summarizes p-values for t-tests of average normalized gaze fixation duration for different pairs of action unit verbs while the upper right triangle represents p-values for t-tests with regards to saccade size.

statistical tests.

We conducted two ANOVA tests with a significance level of $\alpha = 0.05$. One test compared the distributions of gaze fixation duration across the six action unit verb groups while the other test compared the distributions of saccade size. In both cases, the ANOVA resulted in p ¡ 0.001. Thus, post hoc pairwise t-tests were conducted to identify which verb groups were significantly different (Table 2). A Bonferroni correction was additionally applied (α = 0.05/k, where k = 15, the total number of pairwise comparisons) to avoid type I errors when performing the post hoc pairwise comparisons. It was found that the average gaze fixation durations for "pour" and "stir" were significantly greater than those of other verbs (Figure 4A). Saccade sizes for "move" and "stir" were significantly different from those of other verbs (Figure 4B). Saccade sizes for "move" were significantly larger than those of

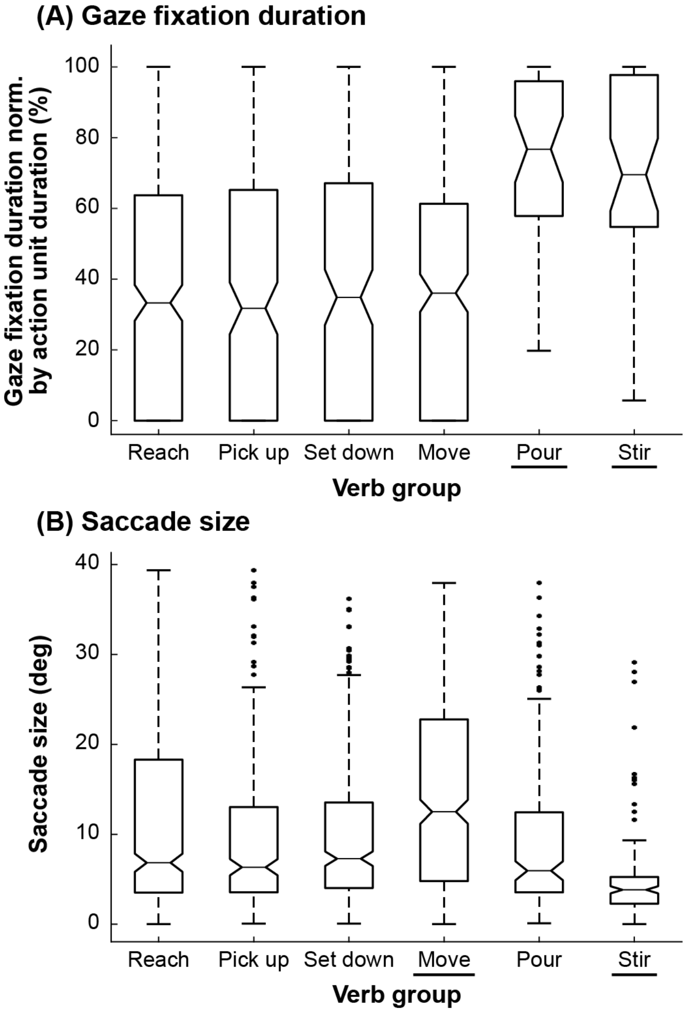


Figure 2.4: Box and whisker plots are shown for each of the six action unit verb groups for (A) normalized gaze fixation duration and (B) saccade size. The tapered neck of each box marks the median while the top and bottom edges mark the first and third quantiles. The whiskers extend to the most extreme data points that are not considered outliers (black dots). For normalized gaze fixation duration, both "pour" and "stir" were statistically significantly different from the other action unit verb groups, as indicated by underlines. For saccade size, both "move" and "stir" were statistically significantly different from the other action unit verb groups.

other verbs while those for “stir” were significantly smaller (Figure 5).

2.4.2 3D Gaze Saliency Maps and Gaze Object Percentages

The 3D gaze saliency map for each object is shown for each of the six subtasks in Figure 5. We use “gaze object” to refer to the object that is intersected by the reconstructed 3D gaze vector. This 3D approach is analogous to the use of 2D egocentric camera videos to identify the gaze object defined as the “object being fixated by eyes” or the “visually attended object” [17]. In the case that multiple objects were intersected by the same gaze vector, we selected the closest object to the subject as the gaze object. We defined the gaze object percentage as the amount of time, expressed as a percent of a subtask, that an object was intersected by a gaze vector. Gaze object percentages, averaged across all 11 subjects, are presented for each of the six subtasks in pie chart form (Figure 5). Although the table in the experiment setup was never manipulated, during some subtasks, the gaze object percentage for the table exceeded 20% for subtasks that included action units related to “set down.”

2.4.3 Recognition of Subtasks Based on Gaze Object Sequences

2.4.3.1 The Gaze Object Sequence

In order to leverage information about the identity of gaze objects in concert with the sequence in which gaze objects were visually regarded, we quantified the gaze object sequence for use in the automated recognition of subtasks. The concept of a gaze object sequence has been implemented previously for human action recognition, but in a different way. [17] performed action recognition with a dynamic Bayesian network having four hidden nodes and four observation nodes. One of the hidden nodes was the true gaze object and one of

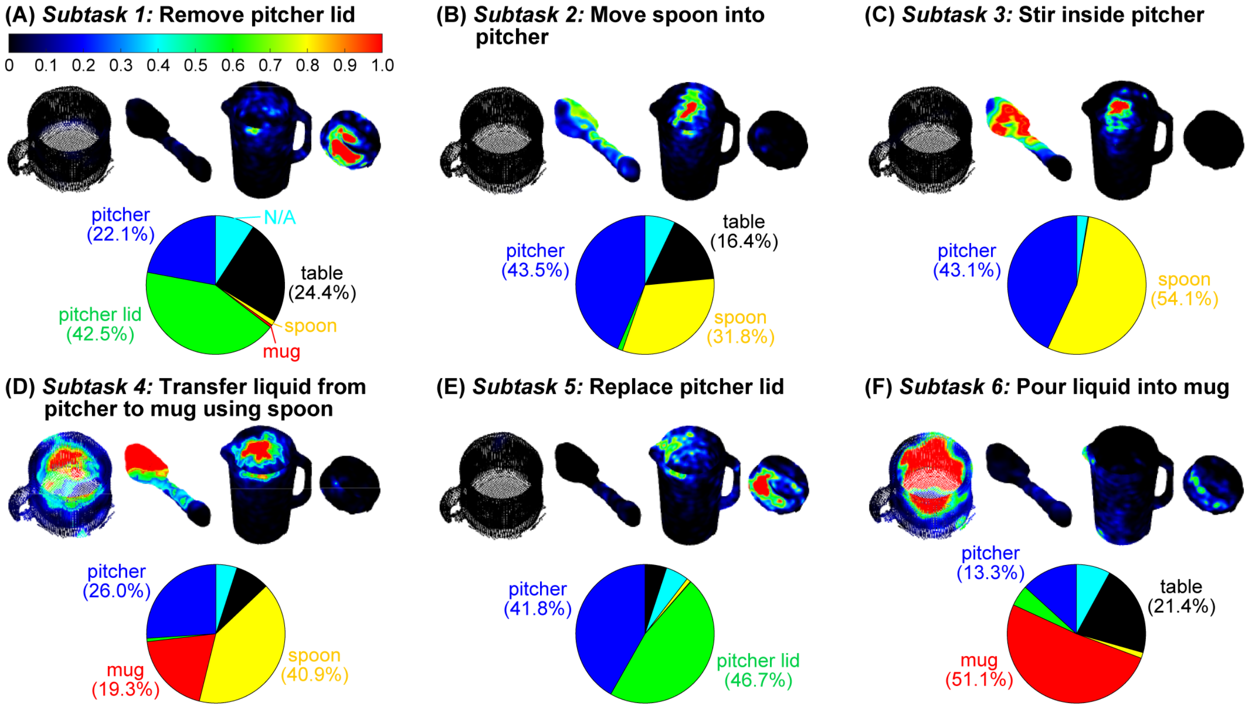


Figure 2.5: Three-dimensional gaze saliency maps of the task-related objects (mug, spoon, pitcher, and pitcher lid) are shown for each of the six subtasks (A–F). The RGB color maps were summed across subjects and then normalized to the [0, 1] range for each subtask. The RGB color scale for all gaze saliency maps is shown in panel (A). Gaze object percentages are reported via pie charts. The colors in the pie charts correspond to the color-coded objects in Figure 1C.

the observation nodes was the estimated gaze object extracted from 2D egocentric camera videos. In this work, we define the gaze object sequence as being comprised of an (M × N) matrix, where M is the number of objects involved in the manipulation task and N is the total number of instances (frames sampled at 60 Hz) that at least one of the M objects was visually regarded, whether through gaze fixation or saccade (Figure 6C). Each of the M = 5 rows corresponds to a specific object. Each of the N columns indicates the number of times each object was visually regarded within a sliding window consisting of 10 frames (Figures 6A,B).

A sliding window was used to filter the raw gaze object sequence to alleviate abrupt changes of values in the matrix. The size of the sliding window was heuristically selected to be large enough to smooth abrupt changes in the object sequence that could be considered as noise, but also small enough so as not to disregard major events within its duration. In preliminary analyses, this sliding window filtration step was observed to improve recognition accuracy.

2.4.3.2 Creating a Library of Characteristic Gaze Object Sequences

Intra- and inter-subject variability necessitate analyses of human subject data that account for variations in movement speed and style. In particular, for pairs of gaze object sequences having different lengths, the data must be optimally time-shifted and stretched prior to comparative analyses. For this task, we used dynamic time warping (DTW), a technique that has been widely used for pattern recognition of human motion, such as gait recognition [50] and gesture recognition [51].

Dynamic time warping compares two time-dependent sequences X and Y, where $X \in \mathbb{R}^{S\times U}$ and $Y \in \mathbb{R}^{S\times V}$. A warping path $W_i = [p_{i1}, p_{i2}, ..., p_{ij}, ..., p_{iK_i}]$ defines an alignment

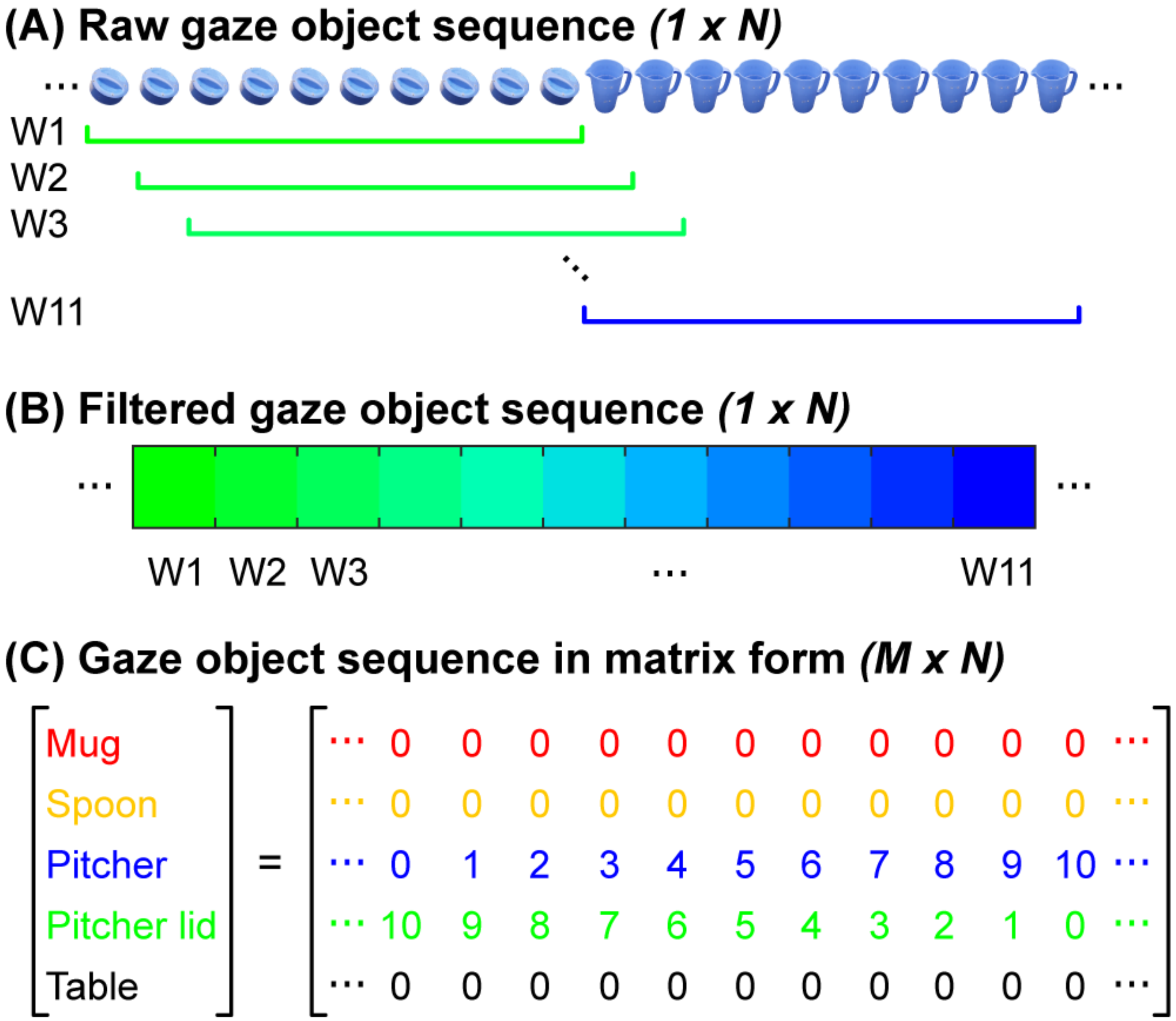


Figure 2.6: (A) Each raw gaze object sequence was represented by a (1 × N) set of frames. In this example, the gaze object transitioned from the pitcher lid to the pitcher. The colors in the figure correspond to the color-coded objects in Figure 1C. (B) The raw sequence of gaze objects was filtered using a rolling window of 10 frames. (C) The gaze object sequence was represented by an (M × N) matrix for M task-relevant objects.

between pairs of elements in X and Y by matching element(s) of X to element(s) of Y. For example, $p_{ij} = (u, v)$ represents the matched pair of x_u and y_v. If the warping path is optimized to yield the lowest sum of Euclidean distances between the two sequences, the DTW distance between the two sequences X and Y can be defined as the following:

$$DTW(X, Y) = \min_{W_i}\{d(W_i) \,|\, W_i \in \langle W_1, W_2, ..., W_L\rangle\}, \tag{2.1}$$

where $d(W_i) = \sum_{j=1}^{K_i} \langle p_{ij}\rangle$ and $\langle p_{ij}\rangle = \|x_u - y_v\|_2$.

In order to identify a characteristic gaze object sequence for each subtask, we employed a global averaging method called dynamic time warping barycenter averaging (DBA), which performs the DTW and averaging processes simultaneously. This method uses optimization to iteratively refine a DBA (average) sequence until it yields the smallest DTW Euclidean distance (see Recognition of Subtasks Using DTW Euclidean Distances) with respect to each of the input sequences being averaged ([52]). The gaze object sequences were averaged across all trials for all subjects for each subtask using an open source MATLAB function provided by the creators of the DBA process ([52]). A total of 43 trials (4 repetitions per each of 11 subjects, less 1 incomplete trial) were available for each subtask. Figure 7 shows visual representations of the DBA gaze object sequence for each of the six subtasks.

2.4.3.3 Recognition of Subtasks Using DTW Euclidean Distances

Traditionally, the Euclidean distance is used as a metric for similarity between two vectors. However, the Euclidean distance alone is not an accurate measure of similarity for time series data ([52]). Here, we use the “DTW Euclidean distance,” which is calculated as the sum of the Euclidean distances between corresponding points of two sequences. The DTW

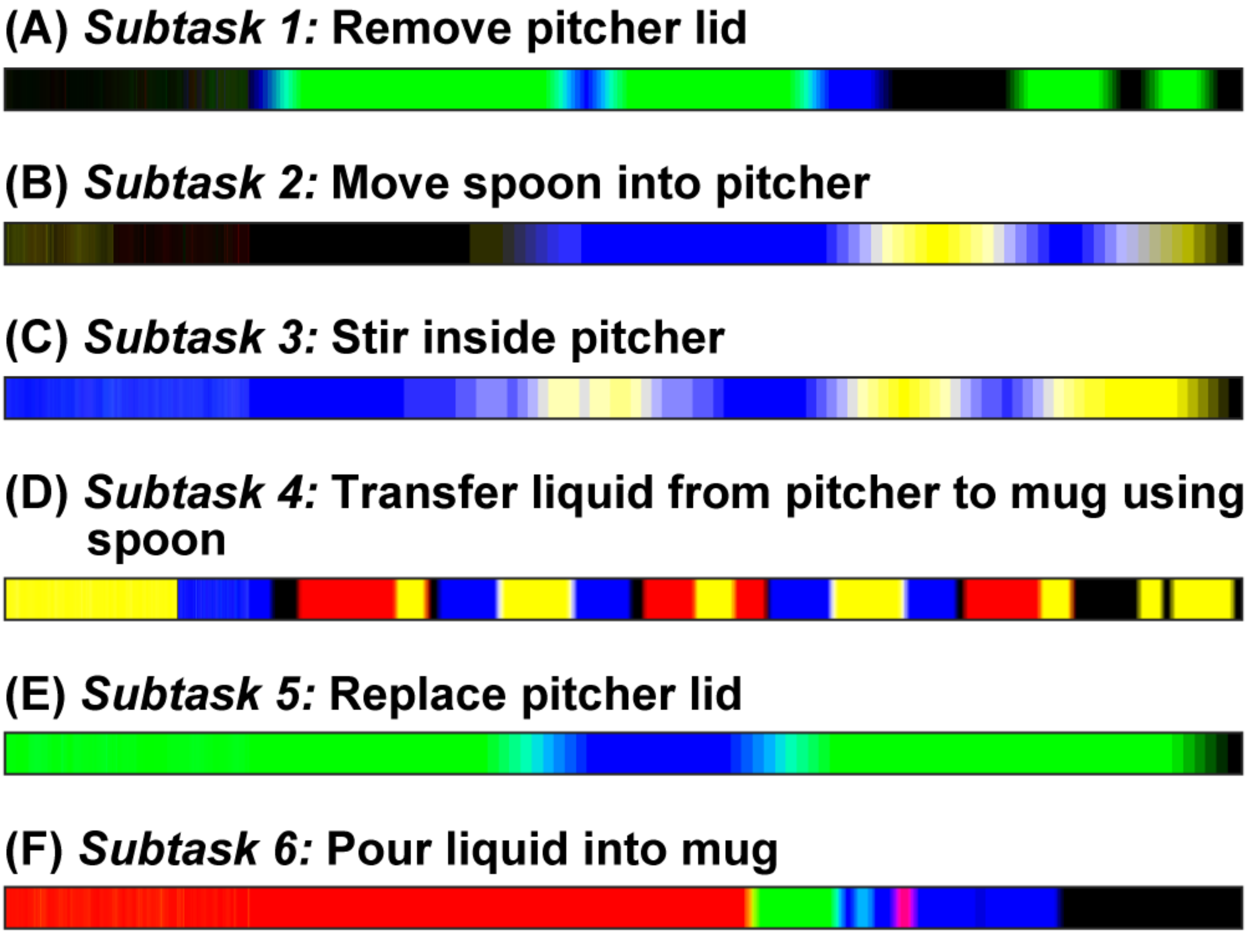


Figure 2.7: Characteristic gaze object sequences were produced using dynamic time warping barycenter averaging over data from 11 subjects for each of six subtasks (A–F). The colors in the figure correspond to the color-coded objects in Figure 1C. The lengths of the sequences were normalized for visualization.

process minimizes the sum of the Euclidean distances, which enables a fair comparison of two sequences. The smaller the DTW Euclidean distance, the greater the similarity between the two sequences. A simple way to associate a novel gaze object sequence with a specific subtask is to first calculate the DTW Euclidean distance between the novel sequence and a characteristic sequence (generated using the DBA process) for each of the six candidate subtasks and to then select the subtask label that results in the smallest DTW Euclidean distance.

Figure 8 shows a novel gaze object sequence and its DTW Euclidean distance with respect to each of the candidate DBA sequences (one for each of six subtasks). The DTW Euclidean distance is reported as a function of the (equal) elapsed times for the novel and DBA gaze object sequences. This enables us to relate recognition accuracy to the percent of a subtask that has elapsed and to comment on the feasibility of real-time action recognition. For instance, for Subtask 4 ("transfer water from pitcher to mug using spoon"), the DTW Euclidean distance between the novel gaze object sequence and the correct candidate DBA sequence does not clearly separate itself from the other five DTW distances until 30% of the novel gaze object sequence has elapsed for the specific case shown (Figure 8). Subtask recognition accuracy generally increases as the elapsed sequence time increases. Figure 8 illustrates how a primitive action recognition approach could be used to label a subtask based on a gaze object sequence alone. However, only one representative novel gaze object sequence was shown as an example.

In order to address the accuracy of the approach as applied to all 43 gaze object sequences, we used a leave-one-out approach. First, one gaze object sequence was treated as an unlabeled, novel sequence. Dynamic time warping barycenter averaging was applied to the remaining sequences. The DTW Euclidean distance was calculated between the novel

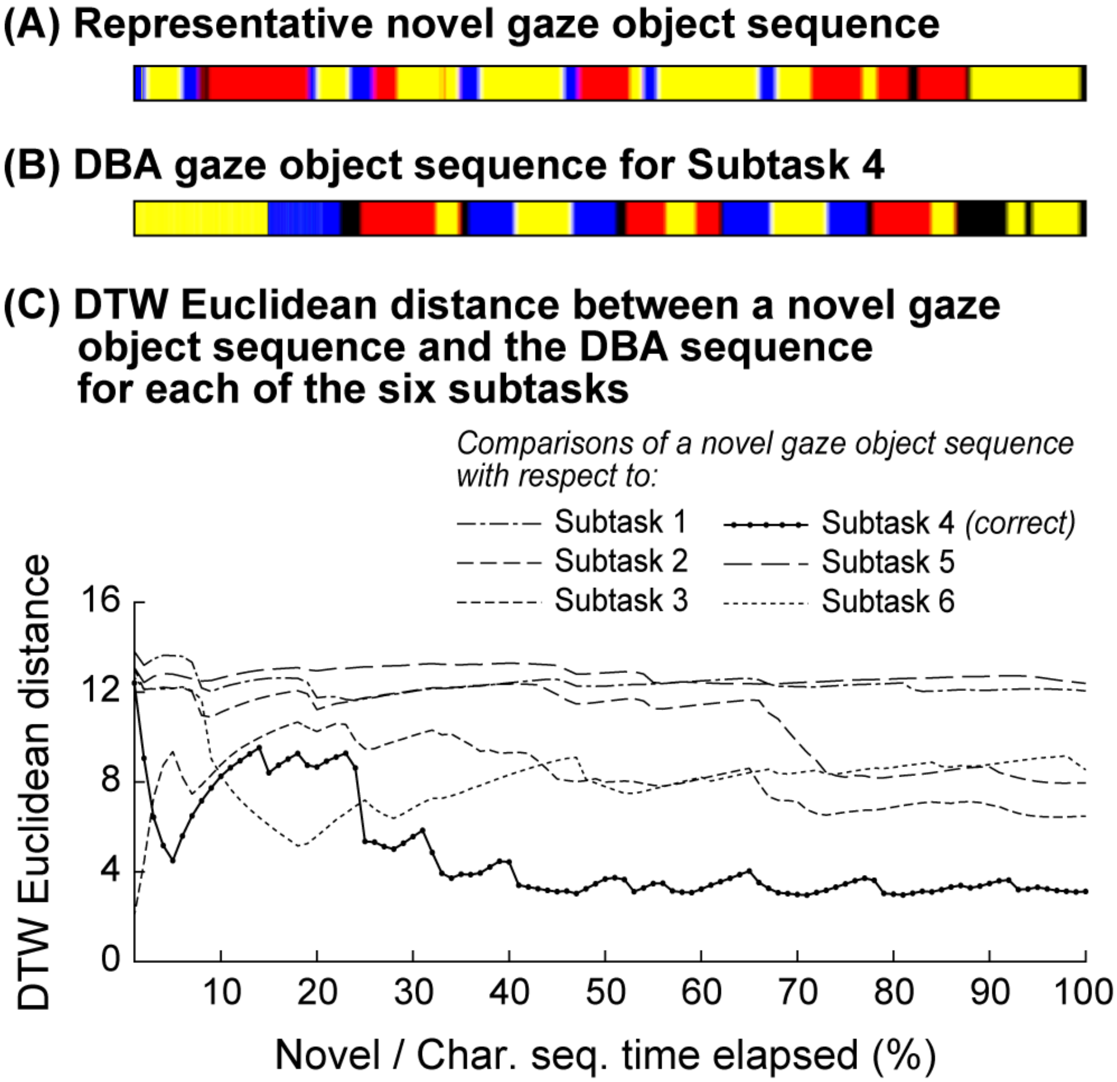


Figure 2.8: (A) A representative novel gaze object sequence is shown. The colors in the figure correspond to the color-coded objects in Figure 1C. (B) A DBA gaze object sequence is shown for Subtask 4, which is the correct subtask label for the novel gaze object sequence shown in panel (A). (C) The DTW Euclidean distance is shown for the comparisons of a novel gaze object sequence and the DBA sequence for each of the six subtasks. The DTW distance was calculated using equal elapsed times for the novel and DBA sequences. The lowest DTW distance would be used to apply a subtask label. Subtask recognition accuracy generally increases as the elapsed sequence time increases.

and candidate DBA sequences, and the pair with the smallest DTW distance was used to label the novel sequence. This process was repeated for each of the gaze object sequences. The DTW distance was calculated using equal elapsed times for the novel and DBA sequences.

The resulting recognition accuracy, precision, and recall for each subtask are reported in Figure 9 as a function of the percent of the subtask that has elapsed. Accuracy represents the fraction of sequences that are correctly labeled. Precision represents the fraction of identified sequences that are relevant to Subtask i. Recall represents the fraction of relevant sequences that are identified ([53])

$$accuracy_i = \frac{TP_i + TNi}{TPi + TNi + FPi + FNi}, \tag{2.2}$$

$$precision_i = \frac{TP_i}{TPi + FPi}, \tag{2.3}$$

$$recall_i = \frac{TP_i}{TPi + FNi}. \tag{2.4}$$

TP_i, TN_i, FP_i, and FN_i represent the number of true positive, true negative, false positive, and false negative sequences when attempting to identify all sequences associated with Subtask i. For example, consider the task of identifying the 43 sequences relevant to Subtask 1 out of the total of (43*6) unlabeled sequences. Using all sequence data, at 100% elapsed time of a novel gaze object sequence, the classifier correctly labeled 36 of the 43 relevant sequences as Subtask 1, but also labeled 10 of the (43*5) irrelevant sequences as Subtask 1. In this case, $TP_1 = 36, TN_1 = 205, FP_1 = 10$, and $FN_1 = 7$. Using Eqs 2–4, this results in an accuracy of 93.4%, precision of 78.2%, and recall of 83.7% for Subtask 1, as shown in Figure 9A.

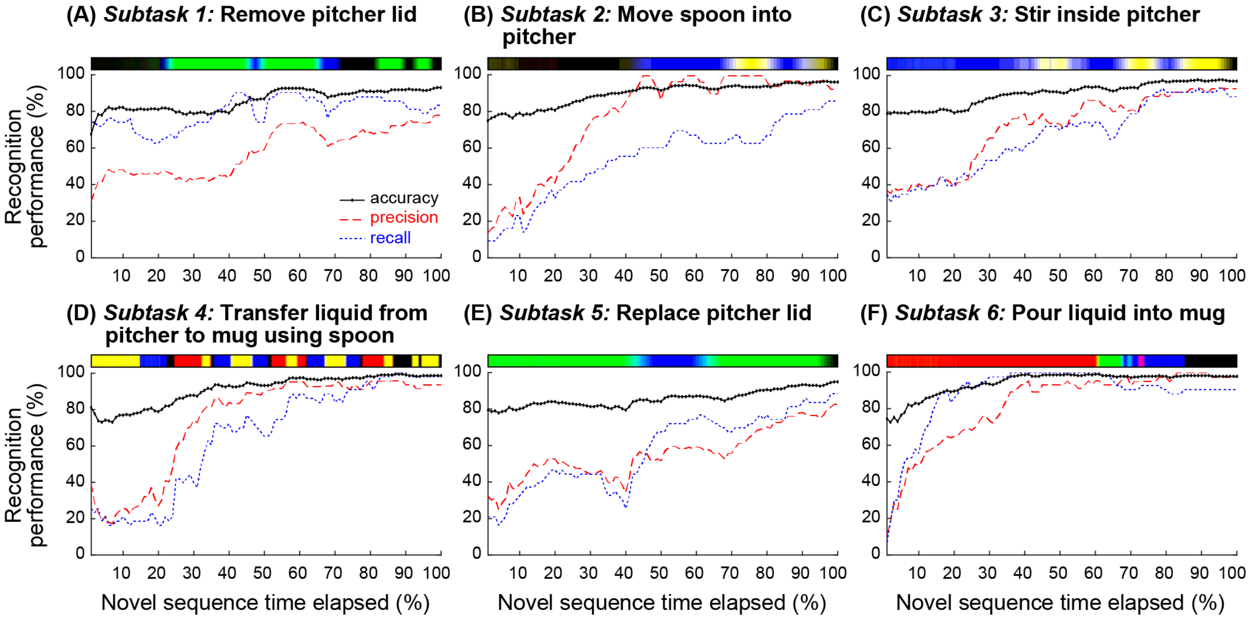
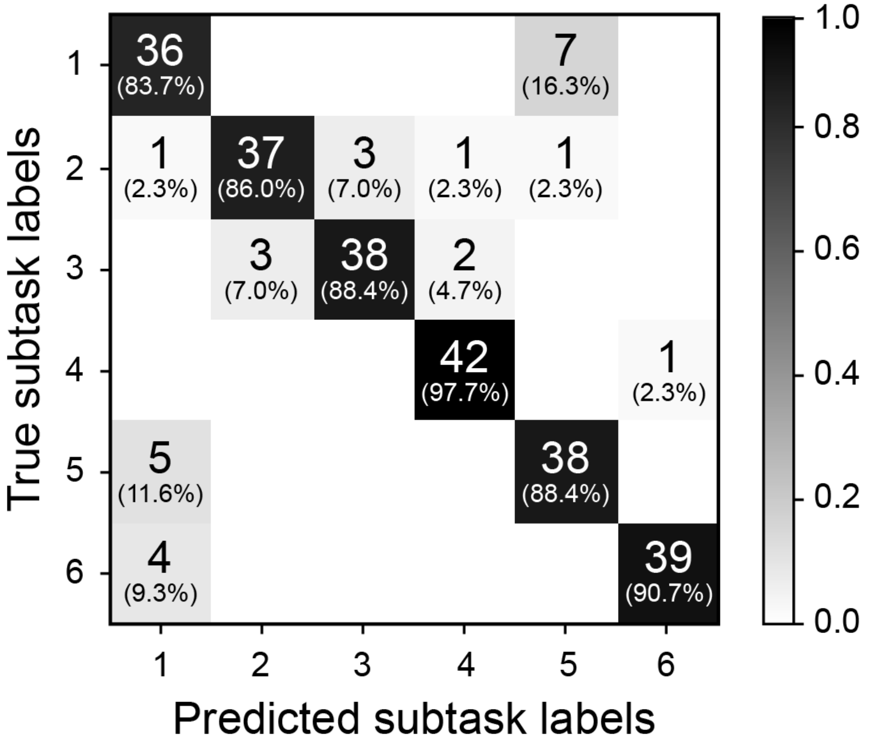


Figure 2.9: Using a leave-one-out approach, the performance of the action recognition algorithm is reported as a function of the elapsed time of a novel gaze object sequence for each subtask. Accuracy (black solid line), precision (red dashed line), and recall (blue dotted line) are shown for each of the six subtasks (A–F). The characteristic gaze object sequence is shown above each subplot. The colors in the sequence correspond to the objects shown in Figure 1C.



	1	2	3	4	5	6
1	36 (83.7%)				7 (16.3%)	
2	1 (2.3%)	37 (86.0%)	3 (7.0%)	1 (2.3%)	1 (2.3%)	
3		3 (7.0%)	38 (88.4%)	2 (4.7%)		
4				42 (97.7%)		1 (2.3%)
5	5 (11.6%)				38 (88.4%)	
6	4 (9.3%)					39 (90.7%)

Figure 2.10: The confusion matrix is shown for 100% of the elapsed time of a novel gaze object sequence for each subtask. Predicted subtask labels (columns) are compared to the true subtask labels (rows). Each subtask has a total of 43 relevant sequences and (43*5) irrelevant sequences. Each shaded box lists the number of label instances and parenthetically lists the percentage of those instances out of 43 relevant subtasks.

Figure 10 shows a confusion matrix that summarizes the subtask labeling performance of our simple action recognition algorithm at 100% of the elapsed time for the novel and DBA gaze object sequences. Predictions of subtask labels (columns) are compared to the true subtask labels (rows). Consider again the task of identifying the 43 sequences relevant to Subtask 1. TP_1 is shown as the first diagonal element in the confusion matrix (row 1, column 1). FP_1 and FN_1 are the sum of off-diagonal elements in the first column and first row, respectively.

2.4.4 Discussion

2.4.4.1 Gaze Fixation Duration and Saccade Size May Reflect Differences in Visual Attention

Eye movements were investigated at the action unit level through gaze fixation duration and saccade size. For gaze fixation duration, both “pour” and “stir” were statistically significantly different from the other action unit verb groups (Figure 4A). The median normalized gaze fixation duration values for “pour” and “stir” were, respectively, 41 and 33% greater than the largest median duration value of the “reach,” “pick up,” “set down,” and “move” verb groups (36% for “move”). The lengthier gaze fixation durations could be due to the fact that pouring and stirring simply took longer than the other movements. The trends could also indicate that more visual attention is required for successful performance of pouring and stirring. For instance, pouring without spilling and stirring without splashing might require greater manipulation accuracy than reaching, picking up, setting down, or moving an object. However, based on the data collected, it is unknown whether subjects were actively processing visual information during these fixation periods. Gaze fixation durations could also be affected by object properties, such as size, geometry, color, novelty, etc. For instance, fixation durations might be longer for objects that are fragile, expensive, or sharp as compared to those for objects that are durable, cheap, or blunt. The effects of object properties on gaze fixation duration and saccade size require further investigation.

For saccade size, both “move” and “stir” were statistically significantly different from the other action unit verb groups (Figure 4B). The relatively large saccade size for “move” was likely a function of the distance by which the manipulated objects were moved during the experimental task. The relatively small saccade size for “stir” (4.7° ± 2.7°) could be due to the small region associated with the act of stirring within a pitcher and the fact that subjects did not follow the cyclic movements of the spoon with their gaze during stirring.

The concept of "quiet eye," originally introduced in the literature with regards to the cognitive behaviors of elite athletes, has been used to differentiate between expert and novice surgeons [54]. Quiet eye has been defined as "the final fixation or tracking gaze that is located on a specific location or object in the visuomotor workspace within 3° of the visual angle for >100 ms" [54]. It has been hypothesized that quiet eye is a reflection of a "slowing down" in cognitive planning (not body movement speed) that occurs when additional attention is paid to a challenging task [55]. Based on the gaze fixation duration trends (Figure 4A), one might hypothesize that pouring and stirring require additional attention. Yet, "stir" was the only verb group that exhibited a small saccade size in the range reported for quiet eye. We are not suggesting that stirring is a special skill that can only be performed by experts; we would not expect a wide range of skill sets to be exhibited in our subject pool for iADL. Nonetheless, it could be reasoned that certain action units may require more visual attention than others and that gaze fixation and saccade size could assist in recognition of such action units employed during everyday tasks.

2.4.4.2 Gaze Saliency Maps Encode Action-Relevant Information at the Subtask and Action Unit Levels

Gaze saliency maps at the subtask level can be used to represent gaze fixation distribution across multiple objects. The gaze saliency maps for the six subtasks (Figure 5) supported Hayhoe and Ballard's finding that gaze fixation during task completion is rarely directed outside of the objects required for the task [56]. Considering Subtask 4, ("transfer water from pitcher to mug using spoon"), the objects comprising the majority of the gaze object percentage pie chart (Figure 5D) were grasped and manipulated (spoon) or were directly affected by an action being performed by a manipulated object (pitcher and mug). While the table was not manipulated, it was often affected by action units that required the picking up or setting down of an object, as for the pitcher lid, spoon, and pitcher in Subtasks 1, 2, and 6 (Figures 5A,B,F), respectively. The gaze fixation percentage for the table was dwarfed by the importance of other objects in Subtasks 4 and 5 (Figures 5D,E).

In some cases, a gaze saliency map could be easily associated with a subtask. For instance, gaze saliency was uniquely, simultaneously intense on the spoon bowl and tip, inner wall of the mug, and inner wall of the pitcher for Subtask 4 ("transfer water from pitcher to mug using spoon") (Figure 5D). In other cases, differences between gaze saliency maps were subtle. For example, the gaze saliency maps were quite similar for the inverse subtasks "remove pitcher lid" and "replace pitcher lid" (Figures 5A,E). In both cases, gaze saliency was focused near the handle of the pitcher lid and the upper rim of the pitcher. However, gaze fixation was slightly more intense near the pitcher spout for Subtask 5 ("replace pitcher lid") because subjects spent time to carefully align the slots in the pitcher lid with the spout for the "pour liquid into mug" Subtask 6 that was to immediately follow.

Likewise, the gaze saliency maps for Subtask 2 ("move spoon into pitcher") and Subtask 3 ("stir inside pitcher") were distinguished only by the subtle difference in gaze fixation distribution on the spoon (Figures 5B,C). The diffuse and homogeneous distribution across the entirety of the spoon for Subtask 2 was contrasted by a focused intensity on the bowl of the spoon for stirring. This was because the "reach for," "pick up," and "move" action units performed with the spoon were summed over time to produce the gaze saliency map at the subtask level. Given that the details of each action unit's unique contribution to the saliency map becomes blurred by temporal summation, it is worth considering gaze saliency maps

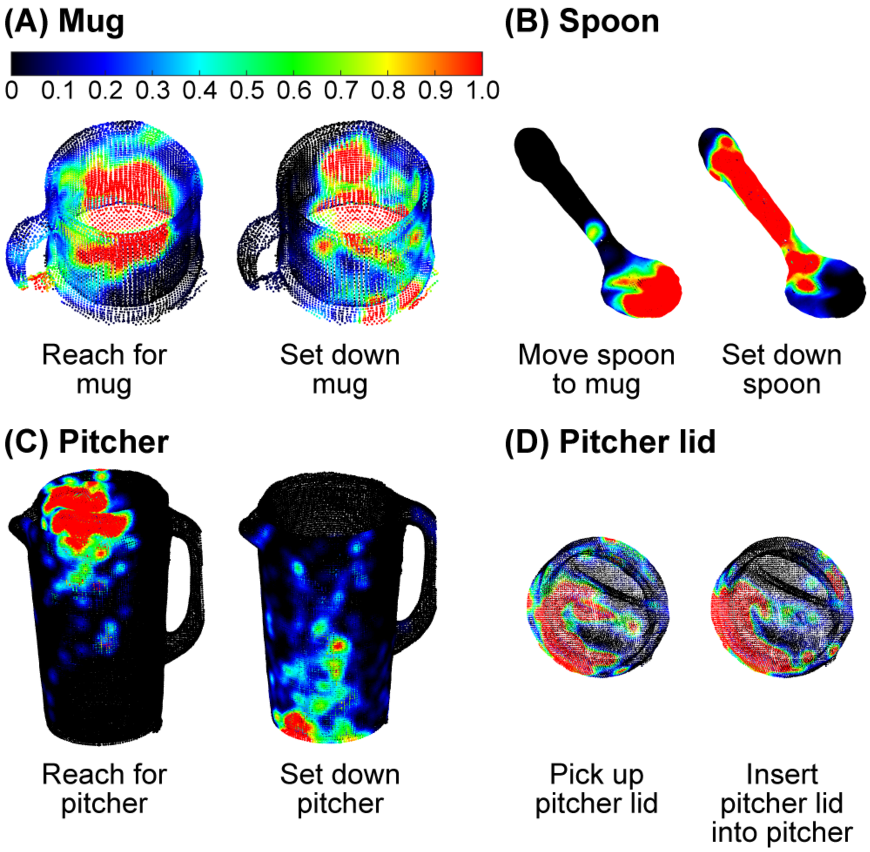


Figure 2.11: Three-dimensional gaze saliency maps of the task-related objects [mug (A), spoon (B), pitcher (C), and pitcher lid (D)] are shown for a subset of action units. The RGB color scale for all gaze saliency maps is shown in panel (A).

at a finer temporal resolution, at the action unit level. Due to the short duration of action units (approximately 1 s long), the gaze saliency maps at the action unit level only involve one object at a time. A few representative gaze saliency maps for different action units are shown in Figure 11. The RGB color intensity maps were summed across subjects and then normalized to the [0, 1] range, with 0 as black and 1 as red, according to the duration of the action unit.

Some gaze saliency maps could also be easily associated with specific action units. For instance, gaze saliency intensity was greatest at the top of the pitcher for the action unit "reach for pitcher," but greatest at the bottom for "set down pitcher" (Figure 11C). By contrast, the gaze saliency maps for the pitcher lid were similar for action units "pick up pitcher lid" and "insert pitcher lid into pitcher." Subtle differences were observed, such as more focused gaze intensity near the slots in the lid, in preparation for the "pour liquid into mug" Subtask 6 that was to immediately follow. Gaze saliency maps for different action units were also similar for the mug (Figure 11A), possibly due to its aspect ratio. Not only is the mug a relatively small object but also its aspect ratio from the subject's viewpoint is nearly one. During both "reach for mug" and "set down mug," gaze fixation was spread around the mug's centroid. This was surprising, as we had expected increased intensity near the mug's handle or base for the "reach" and "set down" action units, respectively, based on the findings of [40]. There are a couple of possible explanations for this. First, the Belardinelli et al. study was conducted with a 2D computer display and subjects were instructed to mimic manipulative actions. In this work, subjects physically interacted with and manipulated 3D objects. It is also possible that subjects grasped the mug with varying levels of precision based on task requirements (or lack thereof). For instance, a mug can be held by grasping its handle or its cylindrical body. Had the task involved a hot liquid, for example, perhaps subjects would have grasped and fixated their gaze on the handle of the mug for a longer period.

Although 3D gaze saliency maps are not necessarily unique for all subtasks and action units, it is likely that a combination of the gaze saliency maps for a subtask and its constituent action units could provide additional temporal information that would enable recognition of a subtask. While beyond the scope of this work, we propose that a sequence of gaze saliency maps over time could be used for action recognition. The time series approaches presented for the analysis of gaze object sequences could similarly be applied to gaze saliency map sequences.

2.4.4.3 Practical Considerations and Limitations of Gaze Saliency Maps

If the dynamic tracking of 3D gaze saliency maps is to be practically implemented, one must address the high computational expense associated with tracking, accessing, and analyzing dense 3D point clouds. In this work, the 3D point clouds for the spoon and pitcher were comprised of approximately 3,000 and 20,000 points, respectively. At least two practical modifications could be made to the gaze saliency map representation. First, parametric geometric shapes could be substituted for highly detailed point clouds of rigid objects, especially if fine spatial resolution is not critical for action recognition. The use of a geometric shapes could also enable one to analytically solve for the intersection point(s) between the object and gaze vector. Second, gaze fixation can be tracked for a select subset of regions or segments, such as those associated with "object affordances," which describe actions that can be taken with an object [57], or ”grasp affordances," which are defined as "object-gripper relative configurations that lead to successful grasps" [58]. Computational effort could then be focused on regions that are most likely to be task-relevant, such as the spout, rim, handle, and base of a pitcher. Additionally, techniques can be leveraged from computer-based 3D geometric modeling. For example, triangle meshes and implicit surfaces have been used for real-time rendering of animated characters[59]. A similar approach could be used to simplify the 3D point clouds. In addition to tracking the shape and movement of an object, one could track the homogeneous properties (e.g., RGB color associated with gaze fixation duration) of patch elements of surfaces. The spatial resolution of each gaze saliency map could be tuned according to the task-relevant features of the object and reduced to the minimal needs for reliable action recognition.

One limitation of this work is that we cannot comment on the subject's true focal point or whether subjects were actively processing visual information. A gaze vector may pass through multiple objects, or even through materials that are not rigid objects (e.g., a stream of flowing water). We calculated the intersection points between a gaze vector and objects in its path and then treated the closest intersection point to the user as a gaze fixation point. This approach may not work if some of the task-relevant objects are transparent and subjects look through one object to visually attend to a more distant object. In this work, objects sometimes passed through the path of a stationary gaze vector, but may not have been the focus of active visual attention. For example, the gaze saliency map for Subtask 3 ("stir inside pitcher") displayed regions of greater intensity on both the bowl of the spoon and the inner wall of the pitcher (Figure 5C). However, the egocentric camera attached to the eye tracker revealed that the gaze fixation point remained near the water level line in the pitcher. Since the spoon was moved cyclically near the inner wall of the pitcher, in the same region as the surface of the water, the gaze fixation point alternated between the spoon and the pitcher. As a result, both the spoon and pitcher gaze saliency maps were affected. In one case, a subject's gaze fixation point was calculated as being located on the outer wall of the pitcher during stirring. This interesting case highlights the fact that a direct line of sight (e.g., to the spoon, water, or inner pitcher surface) may not be necessary for subtask completion, and mental imagery ("seeing with the mind's eye") may be sufficient [60].

Future work should address methods for enhancing the robustness of action recognition algorithms to occlusions. For example, if a gaze object is briefly occluded by a moving object that passes through the subject's otherwise fixed field of view, an algorithm could be designed to automatically disregard the object as noise to be filtered out. In addition, a more advanced eye tracker and/or calibration process could be leveraged to estimate focal length. Focal length could be combined with 3D gaze vector direction to increase the accuracy of gaze object identification in cases, where the 3D gaze vector intersects multiple objects.

Human gaze behavior "in the wild" will differ to some (as yet unknown) extent as compared to the gaze behavior observed in our laboratory setting. Our use of black curtains and the provision of only task-relevant objects enabled the standardization of the experimental setup across subjects. However, this protocol also unrealistically minimized visual clutter, the presence of novel objects, and distractions to the subject. In a more natural setting, one's gaze vector could intersect with task-irrelevant objects in the scene. This would result in the injection of noise into the gaze object sequence, for example, and could decrease the speed and/or accuracy of action recognition. Probabilistic modeling of the noise could alleviate this challenge.

2.4.4.4 The Gaze Object Sequence Can Be Leveraged for Action Recognition to Advance Human–Robot Collaborations

During everyday activities, eye movements are primarily associated with task-relevant objects [61]. Thus, identification of gaze objects can help to establish a context for specific actions. [19] showed that knowledge of gaze location significantly improves action recognition. However, action recognition accuracy was limited by errors in the extraction of gaze objects from egocentric camera video data (e.g., failing to detect objects or detecting irrelevant objects in the background), and gaze objects were not treated explicitly as features for action recognition. Moreover, model development for gaze-based action recognition is challenging due to the stochastic nature of gaze behavior [25]. Using objects tagged with fiducial markers and gaze data from 2D egocentric cameras, Admoni and Srinivasa presented a probabilistic model for the detection of a goal object based on object distance from the center of gaze fixation. In this work, we propose to leverage 3D gaze tracking information about the identity of gaze objects in concert with the temporal sequence in which gaze objects were visually regarded to improve the speed and accuracy of automated action recognition.

In the context of human-robot collaboration, the gaze object sequence could be used as an intuitive, non-verbal control signal by a human operator. Alternatively, the gaze object sequence could be provided passively to a robot assistant that continuously monitors the state of the human operator and intervenes when the human requires assistance. A robot that could infer human intent could enable more seamless physical interactions and collaborations with human operators. For example, a robot assistant in a space shuttle could hand an astronaut a tool during a repair mission, just as a surgical assistant might provide support during a complicated operation. [62] introduced a probabilistic framework for collaboration between a semi-autonomous robot and human co-worker. For a box assembly task, the robot decided whether to hold a box or to hand over a screwdriver based on the movements of the human worker. As there were multiple objects involved in the task, the integration of the gaze object sequence into the probabilistic model could potentially improve action recognition accuracy and speed.

The practical demonstration of the usefulness of gaze object sequence is most likely to occur first in a relatively structured environment, such as that of a factory setting. Despite the unpredictability of human behavior, there are consistencies on a manufacturing line that suggest the feasibility of the gaze object sequence approach. The number of parts and tools used during manual manufacturing operations are uniform in their size and shape and are also limited in number. Although the speed with which a task is completed may vary, the task itself is repetitive. [63] have demonstrated human–robot collaboration for industrial manipulation tasks for which human reaching motions were predicted to enable robot collaboration without collision in a small-shared workspace. In that work, the robot had access to real-time information about the human collaborator's upper limb kinematics, such as palm and arm joint center positions. Focusing on the safety of human–robot collaboration, [64] developed a framework that uses a collision avoidance strategy to assist human workers performing an assembly task in close proximity with a robot arm. Numerous RGB-D cameras were used to track the location and configuration of humans within the collaborative workspace. The common theme of such approaches is to track human kinematics and infer intent from kinematic data alone. The additional use of the gaze object sequence could infer human intent at an earlier stage and further advance safety and efficiency for similar types of human–robot collaboration tasks.

The gaze object sequence could also be demonstrated in the familiar environment of someone's home if a recognition system were properly trained on commonly used objects, where the objects are typically located (e.g., kitchen vs. bathroom), and how they are used. The performance of household robots will largely depend on their ability to recognize and localize objects, especially in complex scenes [65]. Recognition robustness and latency will be hampered by large quantities of objects, the degree of clutter, and the inclusion of novel objects in the scene. The gaze object sequence could be used to address challenges posed by the presence of numerous objects in the scene. While the combinatorial set of objects and actions could be large, characteristic gaze object sequences for frequently used subject-specific iADLs could be utilized to quickly prune the combinatorial set.

Up to now, we have focused primarily on the task-based aspects of gaze tracking for human–robot collaboration. However, gaze tracking could also provide much needed insight into intangible aspects such as human trust in robot collaborators [66]. Our proposed methods could be used to quantify differences in human gaze behavior with and without robot intervention and could enhance studies on the effects of user familiarity with the robot, human vs. non-human movements, perceived risk of robot failure, etc. Consider, for example, a robot arm that is being used to feed oneself [31]. Such a complicated task requires the safe control of a robot near sensitive areas such as the face and mouth and may also be associated with a sense of urgency on the part of the user. A gaze object sequence could reveal high-frequency transitions between task-relevant objects and the robot arm itself, which could indicate a user's impatience with the robot's movements or possibly a lack of trust in the robot and concerns about safety. As the human–robot collaboration becomes more seamless and safe, the frequency with which the user visually checks the robot arm may decrease. Thus, action recognition algorithms may need to be tuned to inter-subject variability and adapted to intra-subject variability as the beliefs and capabilities of the human operator change over time.

Other potential applications of the gaze object sequence include training and skill assessment. For instance, [67] developed a framework that combines Augmented Reality with an Intelligent Tutoring System to train novices on computer motherboard assembly. Via a head-mounted display, trainees were provided real-time feedback on their performance based on the relative position and orientation of tools and parts during the assembly process. Such a system could be further enhanced by, for example, using an expert's gaze object sequence to cue trainees via augmented reality and draw attention to critical steps in the assembly process or critical regions of interest during an inspection process. Gaze object sequences could also be used to establish a continuum of expertise with which skill level can be quantified and certified. [54] described the concepts of "quiet eye" and "slowing down" observed with surgeons performing thyroid lobectomy surgeries. Interestingly, expert surgeons fixated their gaze on the patient's delicate laryngeal nerve for longer periods than novices when performing "effortful" surgical tasks that required increased attention and cognition. Gaze behavior has also been linked with sight reading expertise in pianists [68]. Gaze fixation duration on single-line melodies was shorter for more skilled sight-readers than less skilled sight-readers.

In short, the gaze object sequence generated from 3D gaze tracking data has been demonstrated as a potentially powerful feature for action recognition. By itself, the gaze object sequence captures high-level spatial and temporal gaze behavior information. Moreover, additional features can be generated from the gaze object sequence. For instance, gaze object percentage can be extracted by counting instances of objects in the gaze object sequence. Gaze fixation duration and saccades from one object to another can be extracted from the gaze object sequence. Even saccades to different regions of the same object could potentially be identified if the resolution of the gaze object sequence were made finer through the use of segmented regions of interest for each object (e.g., spout, handle, top, and base of a pitcher).

2.4.4.5 Practical Considerations and Limitations of Gaze Object Sequences

In this work, we have presented a simple proof-of-concept methods for action recognition using a DTW Euclidean distance metric drawn from comparisons between novel and characteristic gaze object sequences. In the current instantiation, novel and characteristic sequences were compared using the same elapsed time (percentage of the entire sequence) (Figure 8). This approach was convenient for a post hoc study of recognition accuracy as a function of time elapsed. However, in practice, the novel gaze object sequence will roll out in real-time and we will not know a priori what percent of the subtask has elapsed. To address this, we propose the use of parallel threads that calculate the DTW Euclidean distance metric for comparisons of the novel sequence with different portions of each characteristic sequence. For instance, one thread runs a comparison with the first 10% of one characteristic gaze object sequence; another thread runs a comparison for the first 20% of the same characteristic gaze object sequence, etc. Such an approach would also address scenarios in which an individual happens to be performing a subtask faster than the population, whose collective behavior is reflected in each characteristic gaze object sequence. For example, it can be seen that the novel gaze object sequence in Figure 8A has a similar pattern as the characteristic gaze object sequence in Figure 8B. However, the individual subject is initially performing the subtask at a faster rate than the population average. The (yellow, blue, black, red, etc.) pattern occurs within the first 10% of the novel sequence, but does not occur until 30% of the characteristic sequence has elapsed. The delayed recognition of the subtask could be addressed using the multi-thread approach described above Figure 8. To further address the computational expense commonly associated with DTW algorithms, one could implement an "unbounded" version of DTW that improves the method for finding matching sequences, which occur arbitrarily within other sequences [69].

For human-robot collaborations, the earlier that a robot can recognize the intent of the human, the more time the robot will have to plan and correct its actions for safety and efficacy. Thus, practical limitations associated with the computational expense of real-time gaze object sequence recognition must be addressed. At the least, comparisons of a novel sequence unfolding in real-time could be made with a library of characteristic subtask sequences using GPUs and parallel computational threads (one thread for each distinct comparison). The early recognition of a novel subtask is not just advantageous for robot planning and control. The computational expense of DTW increases for longer sequences. Thus, the sooner a novel sequence can be recognized, the less time is spent on calculating the proposed DTW Euclidean metric. Since DTW uses dynamic programming to find the best warping paths, a quadratic computational complexity results. While not implemented in this work, the computational expense of the DTW process could be further reduced by leveraging a generalized time warping technique that temporally aligns multimodal sequences of human motion data while maintaining linear complexity [44].

2.4.4.6 Potential Advancements for a Gaze Object Sequence-Based Action Recognition System

As expected, recognition accuracy increased as more of the novel gaze object sequence was compared with each characteristic gaze object sequence (Figure 9). However, the simple recognition approach presented here is not perfect. Even when an entire novel gaze object sequence is compared with each characteristic gaze object sequence, the approach only achieves an accuracy of 96.4%, precision of 89.5%, and recall of 89.2% averaged across the six subtasks. The confusion matrix (Figure 10) shows which subtasks were confused with one another even after 100% elapsed time. Although the percentage of incorrect subtask label predictions is low, the subtasks that share the same gaze objects have been confused the most. For instance, the Subtask 1 ("remove pitcher lid") and Subtask 5 ("replace pitcher lid") were occasionally confused with one another. It is hypothesized that the training of a sophisticated machine learning classifier could improve the overall accuracy of the recognition results, especially if additional features were provided to the classifier. Potential additional features include quantities extracted from upper limb kinematics and other eye tracker data, such as 3D gaze saliency maps.

As with the processing of any sensor data, there are trade-offs with speed and accuracy in both the spatial and temporal domains. In its current instantiation, the gaze object sequence contains rich temporal information, but at the loss of spatial resolution; entire objects are considered rather than particular regions of objects. By contrast, the 3D gaze saliency map and gaze object percentage contain rich spatial information, but at the loss of temporal resolution due to the convolution of eye tracker data over a lengthy period of time. For practical purposes, we are not suggesting that spatial and temporal resolution should be maximized. In practice, an action recognition system need not be computationally burdened with the processing of individual points in a 3D point cloud or unnecessarily high sampling frequencies. However, one could increase spatial resolution by segmenting objects into affordance-based regions [70], or increase temporal resolution by considering the temporal dynamics of action units rather than subtasks.

While object recognition from 2D egocentric cameras is an important problem, solving this problem was not the focus of the present study. As such, we bypassed challenges of 2D image analysis such as scene segmentation and object recognition, and used a marker-based motion capture system to track each known object in 3D. Data collection was performed in a laboratory setting with expensive eye tracker and motion capture equipment. Nonetheless, the core concepts presented in this work could be applied in non-laboratory settings using low-cost equipment such as consumer-grade eye trackers, Kinect RGB-D cameras, and fiducial markers (e.g., AprilTags and RFID tags).

2.4.5 Conclusion

The long-term objective of the work is to advance human-robot collaboration by (i) facilitating the intuitive, gaze-based control of robots and (ii) enabling robots to recognize human actions, infer human intent, and plan actions that support human goals. To this end, the objective of this study was to identify useful features that can be extracted from 3D gaze behavior and used as inputs to machine learning algorithms for human action recognition. We investigated human gaze behavior and gaze-object interactions in 3D during the performance of a bimanual, iADL: the preparation of a powdered drink. Gaze fixation duration was statistically significantly larger for some action verbs, suggesting that some actions such as pouring and stirring may require increased visual attention for task completion. 3D gaze saliency maps, generated with high spatial resolution for six subtasks, appeared to encode action-relevant information at the subtask and action unit levels. Dynamic time warping barycentric averaging was used to create a population-based set of characteristic gaze object sequences that accounted for intra- and inter-subject variability. The gaze object sequence was then used to demonstrate the feasibility of a simple action recognition algorithm that utilized a DTW Euclidean distance metric. Action recognition results (96.4% accuracy, 89.5% precision, and 89.2% recall averaged over the six subtasks), suggest that the gaze object sequence is a promising feature for action recognition whose impact could be enhanced through the use of sophisticated machine learning classifiers and algorithmic improvements for real-time implementation. Future work includes the development of a comprehensive action recognition algorithm that simultaneously leverages features from 3D gaze–object interactions, upper limb kinematics, and hand–object spatial relationships. Robots capable of robust, real-time recognition of human actions during manipulation tasks could be used to improve quality of life in the home as well as quality of work in industrial environments.

CHAPTER 3

Toward Shared Autonomy Control Schemes for Human-Robot Systems: Action Primitive Recognition Using Eye Gaze Features

This chapter was based on work published in the journal Frontiers in Neurorobotics [71].

3.1 Abstract

The functional independence of individuals with upper limb impairment could be enhanced by teleoperated robots that can assist with activities of daily living. However, robot control is not always intuitive for the operator. In this work, eye gaze was leveraged as a natural way to infer human intent and advance action recognition for shared autonomy control schemes. We introduced a classifier structure for recognizing low-level action primitives that incorporates novel three-dimensional gaze-related features. We defined an action primitive as a triplet comprised of a verb, target object, and hand object. A recurrent neural network was trained to recognize a verb and target object, and was tested on three different activities. For a representative activity (making a powdered drink), the average recognition accuracy was 77% for the verb and 83% for the target object. Using a non-specific approach to classifying and indexing objects in the workspace, we observed a modest level of

generalizability of the action primitive classifier across activities, including those for which the classifier was not trained. The novel input features of gaze object angle and its rate of change were especially useful for accurately recognizing action primitives and reducing the observational latency of the classifier.

3.2 Introduction

Activities of daily living (ADLs) can be challenging for individuals with upper limb impairment. The use of assistive robotic arms is an active area of research, with the aim of increasing an individual's functional independence [72]. However, current assistive robotic arms, such as the Kinova arm and Manus arm, are controlled by joysticks that require operators to frequently switch between several modes for the gripper, including a position mode, an orientation mode, and an open/close mode [73, 74].Users need to operate the arm from the gripper's perspective, in an unintuitive Cartesian coordinate space. Operators would greatly benefit from a control interface with a lower cognitive burden that can accurately and robustly inference human intent.

The long-term objective of this work is to advance shared autonomy control schemes so that individuals with upper limb impairment can more naturally control robots that assist with activities of daily living. Toward this end, the short-term goal of this study is to advance the use of eye gaze for action recognition. Our approach is to develop a neural-network based algorithm that exploits eye gaze-based information to recognize action primitives that could be used as modular, generalizable building blocks for more complex behaviors. We define new gaze-based features and show that they increase recognition accuracy and decrease the observational latency [75] of the classifier.

This article is organized as follows. Section Related Work outlines related work with respect to user interfaces for assistive robot arms and action recognition methods. Section Materials and Methods introduces the experimental protocol and proposed structure of an action primitive recognition model, whose performance is detailed in section Results. Section Discussion addresses the effects of input features on classifier performance and considerations for future real-time implementation. Contributions are summarized in section Conclusion.

3.3 Related Work

3.3.1 User Interfaces for Assistive Robot Arms

Many types of non-verbal user interfaces have been developed for controlling assistive robot arms that rely on a variety of input signals, such as electrocorticographic (ECoG) [76], gestures ([77]), electromyography (EMG) ([78]), and electroencephalography (EEG) [79, 80]. Although ECoG has been mapped to continuous, high-DOF hand and arm motion [81, 82], a disadvantage is that an invasive surgical procedure is required. Gesture-based interfaces often require that operators memorize mappings from specific hand postures to robot behaviors [77, 83, 84], which is not natural. EMG and EEG-based interfaces, although non-invasive and intuitive, require users to don and doff EMG electrodes or an EEG cap, which may be inconvenient and require a daily recalibration.

In this work, we consider eye gaze-based interfaces, which offer a number of advantages. Eye gaze is relatively easy to measure and can be incorporated into a user interface that is non-verbal, non-invasive, and intuitive. In addition, with this type of interface, it may be possible to recognize an operator's intent in advance, as gaze typically precedes hand motions [85].

Numerous studies have reported on the use of eye gaze for robot control. In the early 2000's, the eyetracker was used as a direct substitute for a handheld mouse such that the gaze point on a computer display designates the cursor's position, and blinks function as button clicks ([86, 87]). Since 2015, eye gaze has been used to communicate a 3D target position ([14, 88, 89, 28, 90, 91]) for directing the movement of the robotic end effector. No action recognition was required, as these methods assumed specific actions in advance, such as reach and grasp ([36]), write and draw ([89]), and pick and place ([90]). Recently, eye gaze has been used to recognize an action from an a priori list. For instance, Shafti et al. developed an assistive robotic system that recognized subjects' intended actions (including reach to grasp, reach to drop, and reach to pour) using a finite state machine ([92]).

In this work, we advance the use of eye gaze for action recognition. We believe that eye gaze control of robots is promising due to the non-verbal nature of the interface, the rich information that can be extracted from eye gaze, and the low cognitive burden on the operator during tracking of natural eye movements.

3.3.2 Action Representation and Recognition

Moeslund et al. described human behaviors as a composition of three hierarchical levels: (i) activities, (ii) actions, and (iii) action primitives [93]. At the highest level, activities involve a number of actions and interactions with objects. In turn, each action is comprised of a set of action primitives. For example, the activity "making a cup of tea" is comprised of a series of actions, such as "move the kettle to the stove." This specific action can be further divided into three action primitives: "dominant hand reaches for the kettle," "dominant hand moves the kettle to the stove," and "dominant hand sets down the kettle onto the stove."

A great body of computer vision-based studies has already contributed to the recognition of activities of daily living such as walk, run, wave, eat, and drink [94, 95, 96, 97]. These studies detected joint locations and joint angles as input features from external RGB-D cameras and classified ADLs using algorithms such as hidden Markov models (HMMs) and recurrent neural networks (RNNs).

Other studies leveraged egocentric videos taken by head-mounted cameras or eyetrackers ([16, 17, 18, 19, 21, 20, 22, 98, 23]). Video preprocessing methods necessitated first subtracting the foreground and then detecting human hands and activity-relevant objects. Multiple features related to hands, objects, and gaze were then used as inputs for the action recognition using approaches such as HMMs, neural networks, and support vector machines (SVMs). Hand-related features included hand pose, hand location, relationship between left and right hand, and the optical flow field associated with the hand ([18, 23]). Object-related features included pairwise spatial relationships between objects ([21]), state changes of an object (open vs. closed) ([20]), and the optical flow field associated with objects ([18]). The "visually regarded object," defined by [17] as the object being fixated by the eyes, was widely used as the gaze-related feature ([16, 17, 22]). Some studies additionally extracted features such as color and texture near the visually regarded object ([19, 98]).

Due to several limitations, state-of-the-art action recognition methods cannot be directly applied to the intuitive control of an assistive robot via eye gaze. First, computer vision-based approaches to the automated recognition of ADLs have focused on the activity and action levels according to Moeslund's description of action hierarchy ([93]). Yet, state-of-the-art robots are not sophisticated enough to autonomously plan and perform these high-level behaviors. Second, eye movements are traditionally used to estimate gaze point or gaze object alone ([16, 17, 22]). More work could be done to extract other useful features from spatiotemporal eye gaze data, such as time histories of gaze object angle and gaze object angular speed, which are further described in section Gaze-Related Quantities.

3.4 Materials and Methods

3.4.1 Experimental Set-Up

This study was approved by the UCLA Institutional Review Board. The experimental setup and protocol were previously reported in our prior paper ([30]). Data from 10 subjects are reported [nine males, one female; aged 18–28 years; two pure right-handers, six mixed right-handers, two neutral, per a handedness assessment [99] based on the Edinburgh Handedness Inventory [42].Subjects were instructed to perform three bimanual activities involving everyday objects and actions: make instant coffee, make a powdered drink, and prepare a cleaning sponge (Figure 1). The objects involved in these three activities were selected from the benchmark Yale-CMU-Berkeley (YCB) Object Set [43]. We refer to these objects as activity-relevant objects since they would be grasped and manipulated as subjects performed specific activities.

For Activity 1, subjects removed a pitcher lid, stirred the water in the pitcher, and transferred the water to a mug using two different methods (scooping with a spoon and pouring). For Activity 2, subjects were instructed to remove a coffee can lid, scoop instant coffee mix into a mug, and pour water from a pitcher into the mug. For Activity 3, subjects unscrewed a spray bottle cap, poured water from the bottle into a mug, sprayed the water onto a sponge, and screwed the cap back onto the bottle. In order to standardize the instructions provided to subjects, the experimental procedures were demonstrated via a prerecorded video. Each activity was repeated by the subject four times; the experimental setup was reset prior to each new trial.

A head-mounted eyetracker (ETL-500, ISCAN, Inc., Woburn, MA, USA) was used to track the subject's gaze point at 60 Hz with respect to a built-in egocentric scene camera. Per calibration data, the accuracy and precision of the eyetracker were 1.4 deg and 0.1 deg, respectively. The motion of the YCB objects, eyetracker, and each subject's upper limb were tracked at 100 Hz by six motion capture cameras (T-Series, Vicon, Culver City, CA, USA). A blackout curtain surrounded the subject's field of view in order to minimize visual distractions. A representative experimental trial is shown in Supplementary Video 1.

3.4.2 Gaze-Related Quantities

We extract four types of gaze-related quantities from natural eye movements as subjects performed Activities 1–3. The quantities include the gaze object (GO) ([16, 17, 22]) and gaze object sequence (GOS) ([30]). This section describes how these quantities are defined and constructed. As described in section Input Features for the Action Primitive Recognition Model, these gaze-related quantities are used as inputs to a long-short term memory (LSTM) recurrent neural network in order to recognize action primitives.

The raw data we obtain from the eyetracker is a set of 2D pixel coordinates. The coordinates represent the perspective projection of a subject's gaze point onto the image plane of the eyetracker's egocentric scene camera. In order to convert the 2D pixel coordinate into a 3D gaze vector, we use camera calibration parameters determined using a traditional chessboard calibration procedure ([100]) and the MATLAB Camera Calibration Toolbox ([47]). The 3D gaze vector is constructed by connecting the origin of the egocentric camera frame with the gaze point location in the 2D image plane that is now expressed in the 3D global reference frame.

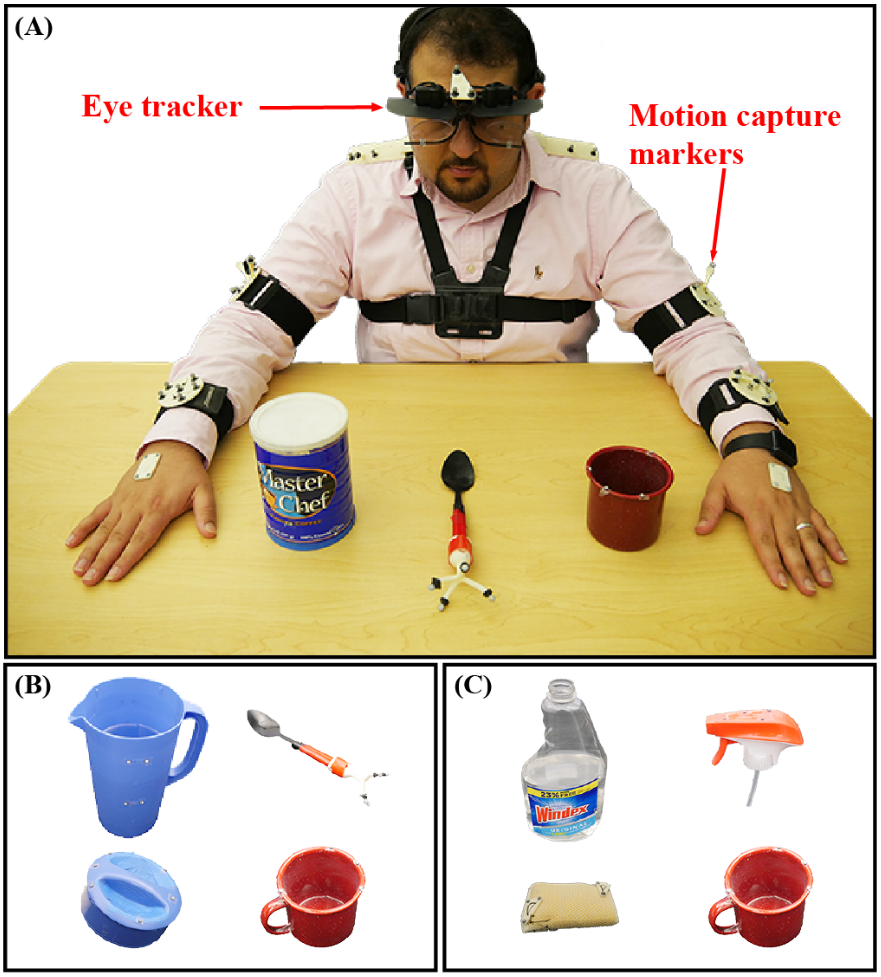


Figure 3.1: (A) A subject prepares to perform Activity 2 (make instant coffee) while eye gaze and kinematics are tracked with a head-mounted eyetracker and motion capture system (not shown). Activity 2 involves a coffee can, spoon and mug. (B) Activity 1 (make a powdered drink) involves a coffee can, spoon and mug. (C) Activity 3 (prepare a cleaning sponge) involves a spray bottle and cap, sponge, and mug. The subject shown in panel (A) has approved of the publication of this image.

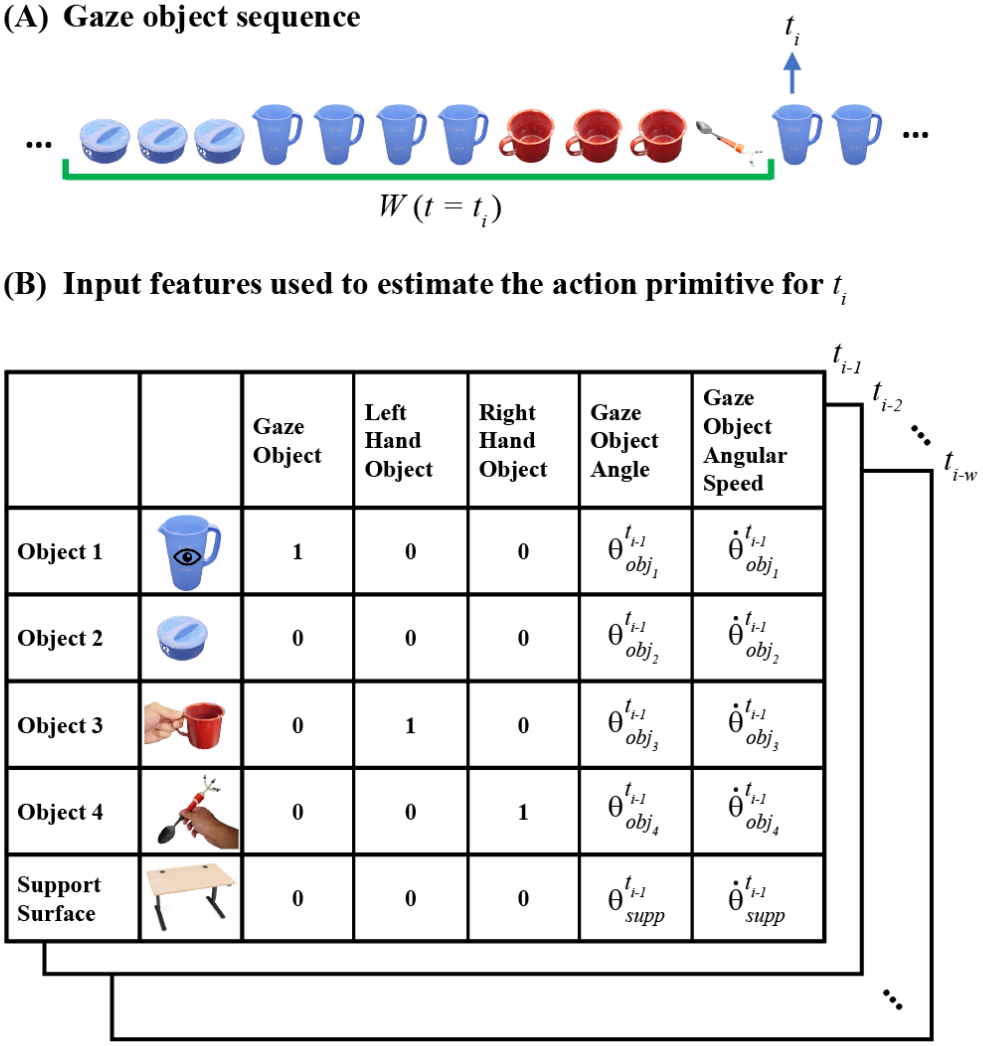
The gaze object (GO) is defined as the first object to be intersected by the 3D gaze vector, as the gaze vector emanates from the subject. Thus, if the gaze vector pierces numerous objects, then the object that is closest to the origin of the 3D gaze vector (within the head-mounted eyetracker) is labeled as the gaze object.

As defined in our prior paper, the gaze object sequence (GOS) refers to the identity of the gaze objects in concert with the sequence in which the gaze objects are visually regarded ([30]). Specifically, the gaze object sequence time history $GOS(t_i)$ is comprised of a sequence of gaze objects sampled at 60 Hz within a given window of time $W(t_i)$ (Figure 2). The time window $W(t_i)$ contains w time steps from t_{i-w} to t_{i-1}.

In this work, we use a value of w = 75 time steps, equivalent to 1.25 s. This time window size was determined from a pilot study whose results are presented in section Effect of Time Window Size on Recognition Accuracy. The pilot study was motivated by the work of Haseeb et al. in which the accuracy of an LSTM RNN was affected by time window size ([101]).

The gaze object angle (GOA) describes the spatial relationship between the gaze vector and each gaze object. The GOA is defined as the angle between the gaze vector and the eye-object vector (Figure 3). The eye-object vector shares the same origin as the gaze vector but ends at an object's center of mass. Each object's center of mass was estimated by averaging the 3D coordinates of the points in the object's point cloud. Each object's point cloud was scanned with a structured-light 3D scanner (Structure Sensor, Occipital, Inc., CA, USA) and custom turntable apparatus. Containers, such as the pitcher and mug, are assumed to be empty for center of mass estimation.

The gaze object angular speed (GOAS) is calculated by taking the time derivative of the GOA. We use the GOAS to measure how the gaze vector moves with respect to other activity-relevant objects. Previously, the gaze object and gaze object sequence have been



		Gaze Object	Left Hand Object	Right Hand Object	Gaze Object Angle	Gaze Object Angular Speed
Object 1		1	0	0	$\theta^{t_{i\text{-}1}}_{obj_1}$	$\dot{\theta}^{t_{i\text{-}1}}_{obj_1}$
Object 2		0	0	0	$\theta^{t_{i\text{-}1}}_{obj_2}$	$\dot{\theta}^{t_{i\text{-}1}}_{obj_2}$
Object 3		0	1	0	$\theta^{t_{i\text{-}1}}_{obj_3}$	$\dot{\theta}^{t_{i\text{-}1}}_{obj_3}$
Object 4		0	0	1	$\theta^{t_{i\text{-}1}}_{obj_4}$	$\dot{\theta}^{t_{i\text{-}1}}_{obj_4}$
Support Surface		0	0	0	$\theta^{t_{i\text{-}1}}_{supp}$	$\dot{\theta}^{t_{i\text{-}1}}_{supp}$

Figure 3.2: (A) The gaze object sequence time history $GOS(t_i)$ within a window of time $W(t_i)$ (green bracket) is shown for Activity 1 (make a powdered drink). (B) To predict the action primitive at time step t_i, input feature vectors (shown as 5×5 matrices for clarity) are created for each of the times from t_{i-w} to t_{i-1}. Activity-relevant objects are sorted according to their frequency of occurrence in the $GOS(t_i)$.

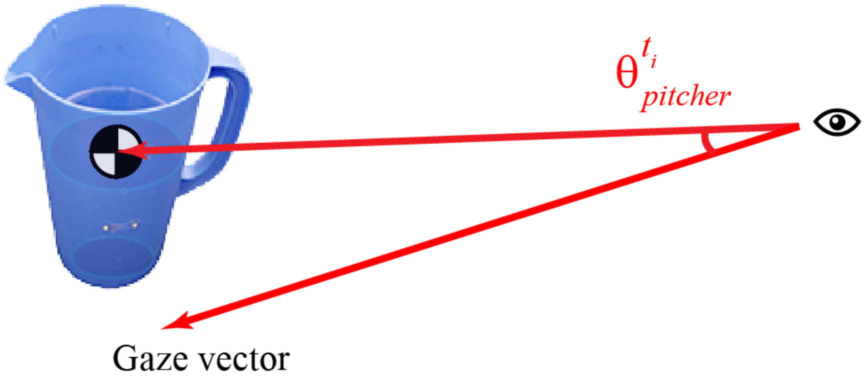


Figure 3.3: Gaze object angle is defined as the angle between the gaze vector and the eye-object vector (ending at the object's center of mass).

used to recognize actions ([17, 22]). To our knowledge, this is the first work to leverage the gaze object angle and gaze object angular speed for action primitive recognition.

3.4.3 Action Primitive Recognition Model

3.4.3.1 Action Primitive Representation

We represent each action primitive as a triplet comprised of a verb, target object (TO), and hand object (HO). Each action primitive can be performed by either the dominant hand or non-dominant hand. When both hands are active at the same time, hand-specific action primitives can occur concurrently.

The verb can be one of four classes: Reach, Move, Set down, or Manipulate. The classes Reach, Move, and Set down describe hand movements toward an object or support surface, with or without an object in the hand. Notably, these verbs are not related to or dependent upon object identity. In contrast, the class Manipulate includes a list of verbs that are highly related to object-specific affordances ([57]). For instance, in Activity 1, the verb "scoop" and “stir” are closely associated with the object “spoon” (Table 1). We refer to these verbs as manipulate-type verbs.

In addition to a verb, the action primitive triplet includes the identity of two objects. The target object TO refers to the object that will be directly affected by verbs such as Reach, Move, Set down, and Manipulate. The hand object HO refers to the object that is currently grasped. For instance, when the dominant hand grasps a spoon and stirs inside a mug, the triplet of the action primitive for the dominant hand is: manipulate (verb), mug (TO), and spoon (HO). A hierarchical description of activities, actions, and action primitives for Activities 1–3 are presented in Table 1.

In order to develop a supervised machine learning model for action primitive recognition, we manually label each time step with the action primitive triplet for either the dominant or non-dominant hand. The label is annotated using video recorded by an egocentric scene camera mounted on the head-worn eyetracker. We annotate each time step with the triplet of a subject’s dominant hand as it is more likely the target of the subject’s attention. For instance, when the dominant hand (holding a spoon) and the non-dominant hand (holding a mug) move toward each other simultaneously, we label the action primitive as “move the spoon to the mug,” where the verb is “move” and the target object is “mug.” However, when the dominant hand is not performing any action primitive, we refer to the non-dominant hand instead. If neither hand is moving or manipulating an object, we exclude that time step from the RNN training process.

3.4.3.2 Input Features for the Action Primitive Recognition Model

Given that the identity of gaze objects will vary across activities, we substitute the specific identities of gaze objects with numerical indices. This is intended to improve the

Activities		Activity 1: make a powdered drink	Activity 2: make instant coffee	Activity 3: prepare a cleaning sponge
Actions		Remove pitcher lid Stir liquid inside pitcher Scoop liquid into mug Pour liquid into mug	Remove coffee can lid Scoop coffee insider can Transfer coffee into mug Stir liquid inside mug	Remove spray bottle cap Transfer cleanser into mug Close spray bottle cap Spray cleanser onto sponge
Action primitives	Verb	Reach, Move, Set down, Manipulate (open, close, stir, scoop, drop pour)	Reach, Move, Set down, Manipulate (open, close, stir, scoop, drop, pour)	Reach, Move, Set down, Manipulate (screw, unscrew, lift, pour, insert, spray)
	TO	Pitcher, pitcher lid, mug, spoon, table	Coffee can, coffee lid, mug, spoon, table	Spray bottle, spray cap, mug, sponge, table
	HO	Pitcher, pitcher lid, mug, spoon	Coffee can, coffee lid, mug, spoon,	Spray bottle, spray cap, cap, mug, sponge

Table 3.1: Each of three activities is divided into actions that are further decomposed into action primitives. Each action primitive is defined as a triplet comprised of a verb, target object (TO), and hand object (HO).

generalizability of our action primitive recognition algorithm across different activities. For each time step t_i, the n activity-relevant objects are sorted in descending order according to their frequency of occurrence in $GOS(t_i)$. Once sorted, the objects are indexed as Object 1 to Object n, such that Object 1 is the object that most frequently appears in the gaze object sequence at t_i. If two or more objects appear in the gaze object sequence with the same frequency, the object with the smaller gaze object angle is assigned the smaller numerical index, as it is aligned most closely to the gaze vector and will be treated preferentially.

Figure 2 exemplifies how activity-relevant objects in a gaze object sequence would be assigned indices at a specific time step t_i. The activity-relevant objects (n = 4) in Activity 1 were sorted according to their frequency of occurrence in $GOS(t_i)$, which is underlined by a green bracket in Figure 2A. Based on frequency of occurrence, the activity-relevant objects were indexed as follows: pitcher (Object 1), pitcher lid (Object 2), mug (Object 3), and spoon (Object 4).

We introduce here the idea of a "support surface," which could be a table, cupboard shelf, etc. In this work, we do not consider the support surface (experiment table) as an activity-relevant object, as it cannot be moved or manipulated and does not directly affect the performance of the activity. Nonetheless, the support surface still plays a key role in the action primitive recognition algorithm due to the strong connection with the verb Set down. In addition, the support surface frequently appears in the GOS.

To predict the action primitive at time step t_i, input feature vectors are created for each of the time steps from time t_{i-w} to t_{i-1}, as shown in Figure 2B. For Activity 1, each input feature vector consists of five features for each of four activity-relevant objects and a support surface. For clarity, each resulting 25×1 feature vector is shown as a five-by-five matrix in Figure 2B. Gaze object, left-hand object, and right-hand object are encoded in the form of one-hot vectors while gaze object angle and angular speed are scalar values.

Gaze object identity was included as an input feature because it supported action recognition in prior studies [16, 17, 22]. We included the hand object as an input feature although it is a component of the action primitive triplet that we seek to recognize. Considering the application of controlling a robotic arm through eye gaze, we expect the robotic system to determine an object's identity before it plans any movements with respect to the object. As a result, we assume that the hand object's identity is always accessible to the classification algorithm. We included the GOA and GOAS as input features because we hypothesized that spatiotemporal relationships between eye gaze and objects would be useful for action primitive recognition. The preprocessing pipeline for the input features is shown in Supplementary Video 1.

3.4.3.3 Action Primitive Recognition Model Architecture

We train a long short-term memory (LSTM) recurrent neural network to recognize the verb and the target object TO for each time step ti. With this supervised learning method, we take as inputs the feature vectors described in section Input Features for the Action Primitive Recognition Model. For the RNN output, we label each time step ti with a pair of elements from a discrete set of verbs and generic, indexed target objects:

$$Verb(t_i) \in V = \{Reach, Move, Setdown, Manipulate\} \tag{3.1}$$

$$TO(t_i) \in O = \{Object_1, Object_2, Object_3, Support\ surface\} \tag{3.2}$$

The target object class Object 4 was excluded from the model output since its usage accounted for ¡1% of the entire dataset. The four verb labels and four TO labels are combined as 16 distinct verb-TO pairs, which are then taken as output classes when we train the RNN.

$$(Verb(t_i, TO(t_i))) \in O \times V = (Reach, Object_1), ..., (Manipulate, Supportsurface) \tag{3.3}$$

As a result, verb-TO pairs that never occur during the training process, such as (Manipulate, Support surface), can be easily eliminated.

In order to evaluate the RNN's performance on the verb and target object individually, we split the verb-TO pairs after recognition. A softmax layer was used as the final layer of the RNN.

$$Verb(t_i) = argmax_{v \in V}(\sum_{o \in O} softmax(Verb(t_i = v, TO(t_i = o))) \tag{3.4}$$

$$TO(t_i) = argmax_{o \in O}(\sum_{v \in V} softmax(Verb(t_i = v, TO(t_i = o))) \tag{3.5}$$

The RNN was comprised of one LSTM layer, three dense layers, and one softmax layer. The LSTM contained 64 neurons and each of the three dense layers contained 30 neurons. The RNN was trained with an Adaptive Momentum Estimation Optimization (Adam), which was used to adapt the parameter learning rate [102]. A dropout rate of 0.3 was applied in order to reduce overfitting and improve model performance. The batch size and epoch number were set as 128 and 20, respectively. The RNN was built using the Keras API in Python with a TensorFlow (version 1.14) backend, and in the development environment of Jupyter Notebook.

Class imbalance is a well-known problem that can result in a classification bias toward the majority class [103]. Since our dataset was drawn from participants naturally performing activities, the training set of samples was not balanced among various verb and TO classes (see sample sizes in Figure 5). An imbalance in TO classes might also result from sorting and indexing the objects as described in section Input Features for the Action Primitive Recognition Model. For instance, Object 1 occurs most frequently in the GOS by definition. Thus, Object 1 is more likely to be the target object than Objects 2 or 3. In order to compensate for the class imbalance, each class' contribution in the cross-entropy loss function was weighted by its corresponding number of samples [104].

The temporal sequence of the target object and verb recognized by the RNN can contain abrupt changes, as shown in the top rows of Figures 5A,B. These abrupt changes occur for limited time instances and make the continuous model prediction unsmooth. Such unstable classifier results might cause an assistive robot to respond unexpectedly. Thus, we implemented a one-dimensional mode filter with an order of m (in our work, m = 12 time steps, equivalent to 0.2 s) to smooth out these sequences [105]:

$$verb(t_i) = mode(\{verb(t_{i-m}), verb(t_{i-m+1}), ..., verb(t_{i-1})\}) \tag{3.6}$$

$$TO(t_i) = mode(\{TO(t_{i-m}), TO(t_{i-m+1}), ..., TO(t_{i-1})\}) \tag{3.7}$$

The sequences after filtering are shown in the middle rows of Figures 5A,B.

Considering that 10 subjects participated in our study, we adopted a leave-one-out cross-validation method. That is, when one subject's data were reserved for testing, the other nine subjects' data were used for training.

3.4.3.4 Performance Metrics for Action Recognition

In order to evaluate the performance of the action primitive classification, we assessed overall accuracy, precision, recall, and the F1-score. Overall accuracy is the number of correctly classified samples divided by the total size of the dataset. For each class of verb or target object, precision represents the fraction of correctly recognized time steps that actually belong to the given class, and recall represents the fraction of the class that are successfully recognized. We use TP, TN, and FP to represent the number of true positives, true negatives, and false positives when classifying a verb or target object class.

$$overall\ accuracy = \frac{\sum TP}{total\ size\ of\ dataset} \tag{3.8}$$

$$precision = \frac{TP}{TP + FP} \tag{3.9}$$

$$recall = \frac{TP}{TP + TN} \tag{3.10}$$

The F1-score is the harmonic mean of precision and recall.

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3.11}$$

We also used performance metrics that were related to the temporal nature of the data. In order to evaluate how early an action primitive was successfully recognized, we adopted the terminology "observational latency," as defined in [75]. The term was defined as “the difference between the time a subject begins the action and the time the classifier classifies the action,” which translates to the amount of time that a correct prediction lags behind the start of an action primitive. It should be noted that the observational latency does not include the computation time that the recognition algorithm requires to preprocess the input data and recognize the actions by the model.

We conservatively judged the success of an action primitive's classification by checking whether more than 75% of its time period was predicted correctly. Summary statistics for observational latency are reported for action primitives that were deemed correct according to this 75% threshold. Observational latency is negative if the action primitive is predicted before it actually begins.

3.5 Results

Recall our aim of specifying the three components of the action primitive triplet: verb, target object, and hand object. Given that the hand object is already known, as described in section Input Features for the Action Primitive Recognition Model, we report on the ability of the RNN to recognize the verb and target object. A demonstration of the trained RNN is included in Supplementary Video 1.

3.5.1 Effect of Time Window Size on Recognition Accuracy

In order to set the time window size, we conducted a pilot study inspired by [101]. We tested how the F1-scores of the verb and TO classes varied as the time window size was increased from five time steps (equivalent to 83 ms) to 2 s in increments of five time steps (Figure 4). Considering the average duration of an action primitive was only 1.2 s, we did not consider time window sizes beyond 2 s.

As seen in Figure 4A, time window size had a more substantial effect on the recognition

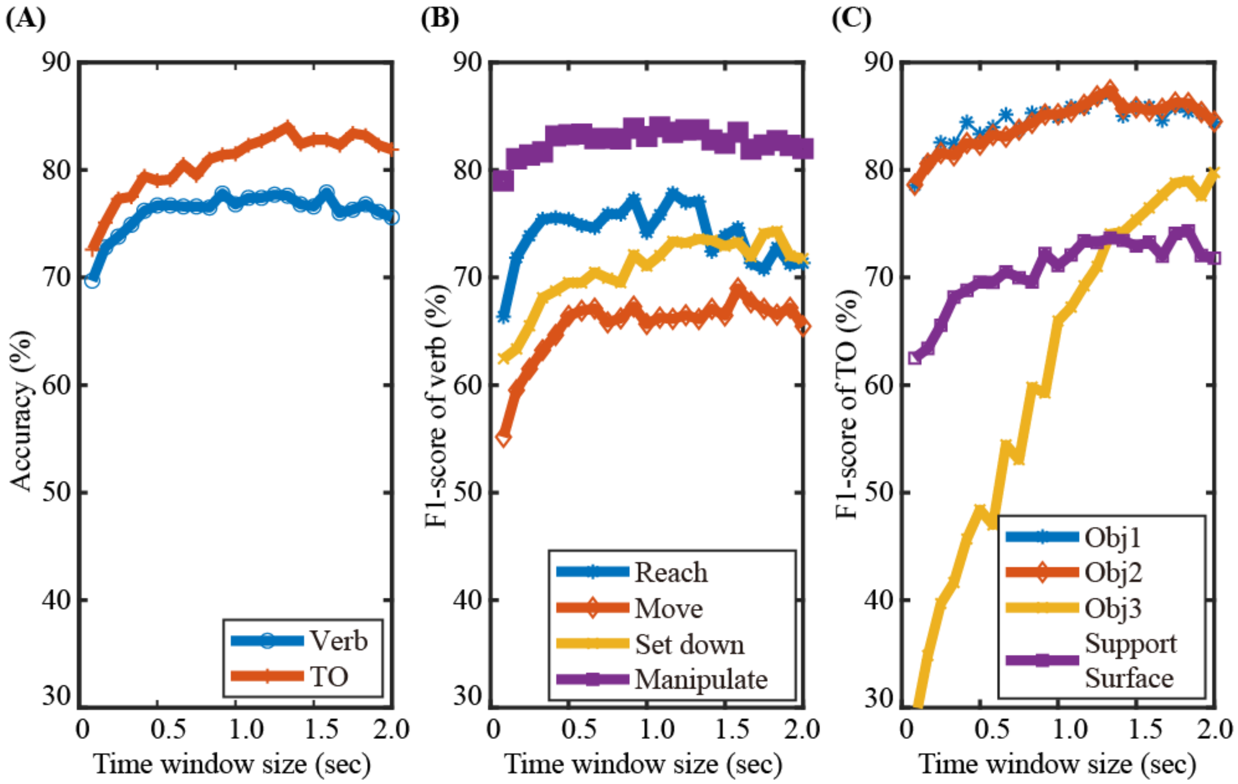
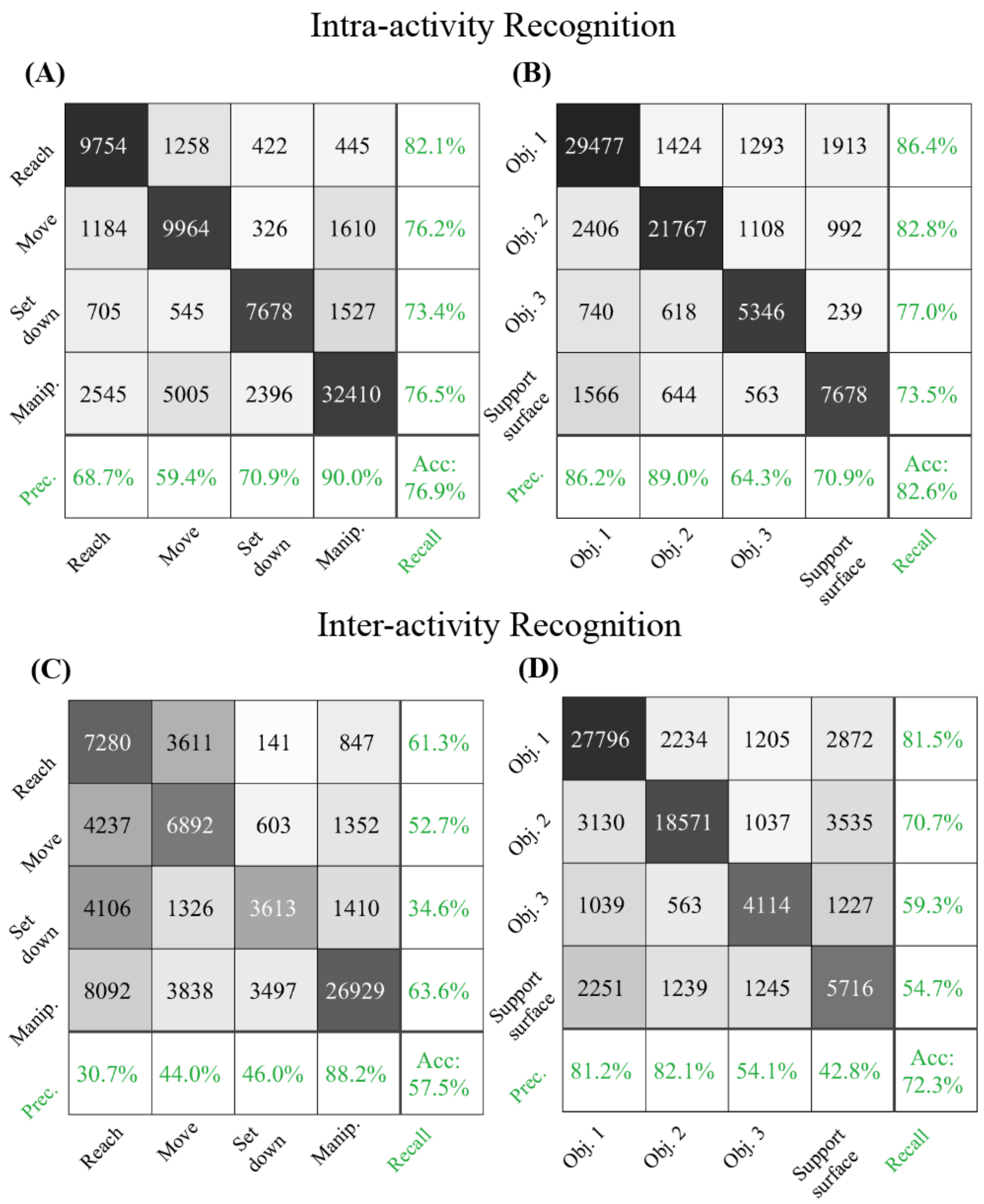


Figure 3.4: The effect of time window size (ranging from 83ms to 2 s) on recognition performance is shown for Activity 1. The overall recognition accuracy for verb and target object are shown in (A). F1-scores for the verb and target object classes are shown in (B,C), respectively.

of TO than that of verb. This is due to the fact that time window size can greatly affect the data sample distributions among target object classes as a result of sorting and indexing the activity-relevant objects. Figure 4C shows that the TO class Object 3 was especially sensitive to the window size. The corresponding F1-score continuously increased from 30% to 80% until the window size reached 1.8 s. Recognition performance of the other three TO classes Object 1, Object 2, and Support surface were also improved as the time-window size was increased from 80 ms to 1.25 s. The increased F1-scores of the TO classes can be partly attributed to alleviated class imbalance problem as the time window was lengthened, especially for the class Object 3. The number of data samples of Object 3 greatly increased due to the nature of sorting and indexing objects according to their frequency of occurrence in gaze object sequence.

As seen in Figure 4B, the F1-scores of the verb classes Reach, Move, and Manipulate increased as the time-window size increased from 80 ms to 0.5 s. Little improvement in the F1-scores was observed for time window sizes $>$ 0.5 s, except for Set down. This suggested that a memory buffer of 0.5 s might be sufficient for predicting the verb class based on eye gaze. Gaze-related information collected long before the start of an action primitive was very likely to be irrelevant to the verb.

Considering the effect of the time window size on the classification accuracy of both the verb and target object (Figure 4), we decided to use a time window size of 1.25 s. A time window longer than 1.25 s might slightly improve recognition performance, but with additional computational cost.



Intra-activity Recognition

(A)

	Reach	Move	Set down	Manip.	Recall
Reach	9754	1258	422	445	82.1%
Move	1184	9964	326	1610	76.2%
Set down	705	545	7678	1527	73.4%
Manip.	2545	5005	2396	32410	76.5%
Prec.	68.7%	59.4%	70.9%	90.0%	Acc: 76.9%

(B)

	Obj. 1	Obj. 2	Obj. 3	Support surface	Recall
Obj. 1	29477	1424	1293	1913	86.4%
Obj. 2	2406	21767	1108	992	82.8%
Obj. 3	740	618	5346	239	77.0%
Support surface	1566	644	563	7678	73.5%
Prec.	86.2%	89.0%	64.3%	70.9%	Acc: 82.6%

Inter-activity Recognition

(C)

	Reach	Move	Set down	Manip.	Recall
Reach	7280	3611	141	847	61.3%
Move	4237	6892	603	1352	52.7%
Set down	4106	1326	3613	1410	34.6%
Manip.	8092	3838	3497	26929	63.6%
Prec.	30.7%	44.0%	46.0%	88.2%	Acc: 57.5%

(D)

	Obj. 1	Obj. 2	Obj. 3	Support surface	Recall
Obj. 1	27796	2234	1205	2872	81.5%
Obj. 2	3130	18571	1037	3535	70.7%
Obj. 3	1039	563	4114	1227	59.3%
Support surface	2251	1239	1245	5716	54.7%
Prec.	81.2%	82.1%	54.1%	42.8%	Acc: 72.3%

Figure 3.5: Intra-activity recognition results for Activity 1 are shown in confusion matrix form for (A) verb and (B) target object. Inter-activity recognition results for an RNN trained on Activity 2 and tested on Activity 1 are shown for (C) verb and (D) target object. Integers in the confusion matrices represent numbers of samples. The confusion matrices are augmented with precision, recall, and accuracy results (green).

3.5.2 Intra-Activity Recognition

We report results for intra-activity recognition, in which we trained and tested the recurrent neural network on the same activity. These results describe how well the RNN recognized novel instances of each activity despite variability inherent to activity repetition. Intra-activity recognition results for Activity 1 are shown in Figure 5 in the traditional form of confusion matrices. The rows correspond to the true class and the columns correspond to the predicted class. For brevity, intra-activity recognition results for Activities 1 and 2 are also shown in Table 2 in the form of F1-scores. The weighted averages of F1-scores for verb and target object were each calculated by taking into account the number of data samples for each class. The RNN was not trained on Activity 3 due to its smaller dataset as compared to Activities 1 and 2. Thus, no intra-activity recognition results were reported for Activity 3.

We augmented the traditional confusion matrix used to report results according to true and predicted classes with additional metrics of precision and recall (Figure 5). Precision and recall were reported as percentages (in green) in the far right column and bottom-most row, respectively. The cell in the lower-right corner represented the overall recognition accuracy.

The data samples were not balanced among various verb and TO classes since our dataset was drawn from participants naturally performing activities. The proportion of each verb and TO class in Activity 1 was the sum of the corresponding row in Figures 5A,B divided by the total size of the dataset (77,774 time step samples). The proportions for the verb classes were 15% for Reach, 17% for Move, 13% for Set down, and 55% for Manipulate. The proportions for the target object classes were 44% for Object 1, 34% for Object 2, 9% for Object 3, and 13% for Support surface.

Intra- or Inter-activity recognition	**Intra**	**Inter**	**Inter**	**Intra**	**Inter**	**Inter**
Activity # (training)	1	1	1	2	2	2
Activity # (testing)	1	2	3	2	1	3
F1-scores for verb recognition (%)						
Reach	74.8	52.9	54.8	56.5	40.9	55.6
Move	66.8	36.6	61.1	59.5	48.0	60.5
Set down	72.1	49.3	45.3	59.6	39.5	44.4
Manipulate	82.7	73.7	72.7	81.4	73.9	71.8
Verb Average	77.4	60.3	63.6	68.6	59.9	63.1
F1-scores for target object recognition (%)						
Object 1	86.3	72.1	78.0	80.2	81.3	77.4
Object 2	85.8	80.7	83.6	87.2	76.0	80.8
Object 3	70.1	41.7	52.5	55.2	56.6	56.8
Support surface	72.2	56.9	49.8	69.3	48.0	46.6
TO Average	82.8	73.0	74.9	81.1	72.8	73.4

Table 3.2: The RNN performance for intra- and inter-activity recognition is reported via F1-scores (%). Weighted averages of F1-scores that account for the number of data samples in each class are reported for both verb and target object (TO).

The RNN achieved a good performance in recognizing the majority verb class Manipulate (precision: 90%, recall: 77%) and the TO class Object 1 (precision: 86%, recall: 86%), which laid a solid foundation for its overall accuracy (verb: 77%, TO: 83%).

3.5.3 Inter-Activity Recognition

We report results for inter-activity recognition, in which we trained and tested the recurrent neural network on different activities. These results describe how well the RNN can recognize verbs and target objects despite variability across different activities. To evaluate the algorithm's cross-activity generalizability, an RNN trained on Activity 2 (make instant coffee) was tested on Activity 1 (make a powdered drink), and vice versa. RNNs trained on Activity 1 and Activity 2 were additionally tested on Activity 3 (prepare a cleaning sponge). The confusion matrices of an RNN trained on Activity 2 and tested on Activity 1 are shown in Figures 5C,D for verb and target object estimation, respectively. For brevity, additional inter-activity recognition results are presented in Table 2 in the form of F1 scores.

We also compared intra-activity and inter-activity performance of RNN models tested on the same activity. For this, we subtracted the average F1-scores for inter-activity recognition from those of the appropriate intra-activity recognition for RNNs tested on Activity 1 and Activity 2. As expected, when testing with an activity that differed from the activity on which the RNN was trained, the classification performance decreased. The average F1-scores of verb and target object each dropped by 8% when the RNN was trained on Activity 1 and tested on Activity 2. The average F1-scores of verb and target object dropped by 18 and 10%, respectively, when the RNN was trained on Activity 2 and tested on Activity 1. The average F1-score decreases were no larger than 20%, which suggested that the classification algorithm was able to generalize across activities to some degree. In addition, despite the

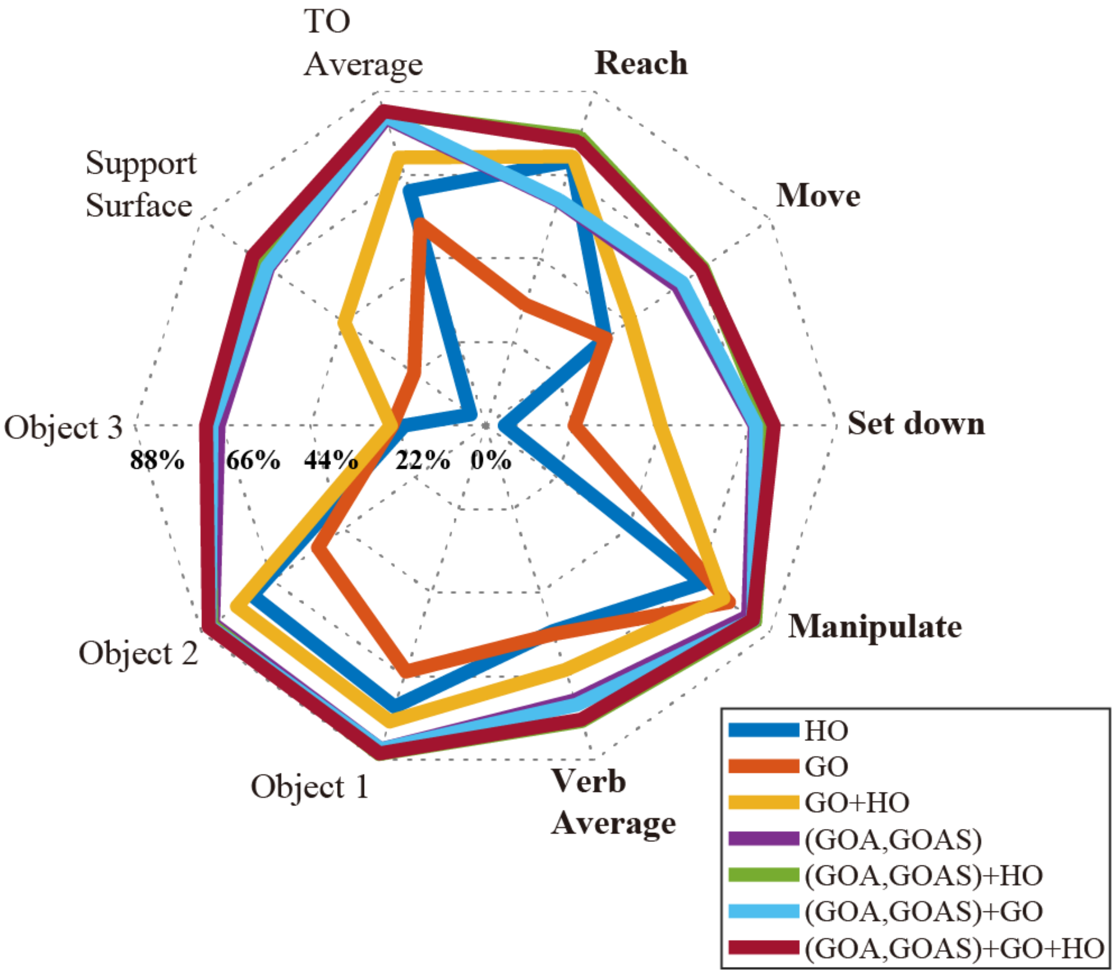


Figure 3.6: For Activity 1, RNN performance is reported by F1-scores for different combinations of input features (HO, GO, GOA, GOAS) using a radar chart. Axes represent the verb (bold) and target object classes. F1-score gridlines are offset by 22%. Each of the polygons corresponds to one combination of input features. The combined use of HO, GO, GOA, and GOAS features resulted in the best performance; HO alone performed the worst.

fact that Activity 3 shared only one common activity-relevant object (mug) with the other two activities, the average F1-scores of verb and TO achieved for Activity 3 were slightly higher than those of the other inter-activity recognition tests (Table 2).

3.5.4 Effect of Input Features on Recognition Accuracy

In order to evaluate feature importance, we compared the classification performance achieved in Activity 1 with various combinations of input features using a radar chart (Figure 6). Axes represented the verb and target object classes. Gridlines marked F1-scores in increments of 22%. Classification using HO alone was poor, with F1-scores for “Set down” and “Object 3” being ¡10%. Only slightly better, classification using GO alone was still not effective, with F1-scores of the “Set down,” “Object 3,” and “Support surface” only reaching values near 22%. In contrast, GOA-based features (GOA, GOAS) alone outperformed both HO and GO on their own in every verb and target object class. With the exception of “Reach,” GOA-based features alone also outperformed the use of HO and GO together.

Although the feature HO alone did not provide good recognition result, it could substantially improve the classification performance when used in concert with GOA-based features. For every class, the F1-scores achieved with the combination of GOA-based feature and HO were equal to or higher than with the GOA-based feature alone.

3.5.5 Effect of Input Features on Observational Latency

The time histories of the verb and target object recognition for a representative Activity 1 trial are shown in Figures 7A,B. In each of Figures 7A,B, the top colorbar represents a time history of raw prediction results. The middle colorbar shows the output of the mode filter that smooths the raw prediction results. The bottom colorbar represents the ground truth. White gaps in the ground truth correspond to instances when neither hand was moving or manipulating an object. The observational latency is obtained by comparing the middle and bottom colorbars.

While Figure 7 shows the observational latency for a single representative trial, the observational latencies for all trials and participants are presented in Figure 8. Specifically, Figures 8A,B, summarize results for the recognition of verb and target object, respectively, for an RNN trained and tested on Activity 1. Figure 8 illustrates the effect of input features on observational latency by comparing the results of an RNN that only used GO and HO as input features to those of an RNN that additionally used GOA, and GOAS as input features.

We hypothesized that the incorporation of GOA-based input features could significantly decrease observational latency. To test this, we conducted a Wilcoxon signed-rank test (following a Lilliefors test for normality) with a total of 714 action primitives. The one-tailed p-values for the verbs and target objects were all less than the α level of 0.05 except for the target object of pitcher lid. Thus, we concluded that the use of GOA and GOAS as input features in addition to GO and HO resulted in a reduction in observational latency (Figure 8).

3.6 Discussion

3.6.1 Features Based on Gaze Object Angle Improve Action Primitive Recognition Accuracy

The long-term objective of this work is to advance shared autonomy control schemes so that individuals with upper limb impairment can more naturally control robots that assist with activities of daily living. One embodiment of such a teleoperated system could include both a joystick and eyetracker as user input devices. The short-term goal of this study was to improve action primitive recognition accuracy and observational latency. We pursued this goal by (i) focusing on the recognition of low-level action primitives, and (ii) defining eye gaze-based input features that improve action primitive recognition.

Previous studies leveraged egocentric videos to recognize actions when a subject was naturally performing ADLs. The features reported in these studies can be divided into three categories: features based on human hands, objects, or human gaze. Examples of hand-based features include hand location, hand pose, and relative location between left and right hands [18, 23]. Fathi et al. relied on changes in the state of objects, such as the state of the "coffee jar" (open vs. closed) [20], to recognize actions. Behera et al. used spatiotemporal relationships between objects as classifier inputs ([21]). Features related to human gaze included the gaze-object, which was widely used to classify actions ([17, 22]). The use of object appearance (histogram of color and texture) in the neighborhood of the gaze point was also effective in improving recognition accuracy ([19, 98]).

Considering the long-term objective of this work, we elected not to rely solely on features based on human hands or objects for action primitive recognition. Features based on human hands are only available when subjects use their own hands to directly grasp and manipulate objects. For the assistive robot application we envision, features of human hands such as hand location, hand pose, and relative location between left and right hands ([18, 23]) will not be available. Features based on objects are consequence of hand motions, such as changes in the states of objects or spatiotemporal relationships between objects. Such object-based features would only be available in hindsight and cannot be collected early enough to be useful for the proposed assistive robot application.

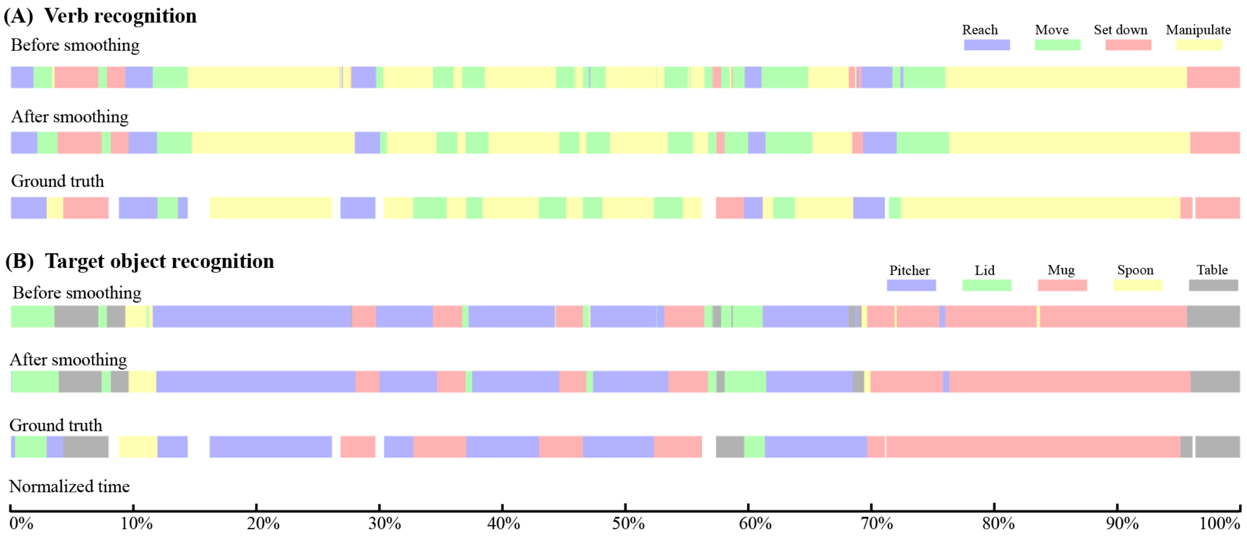
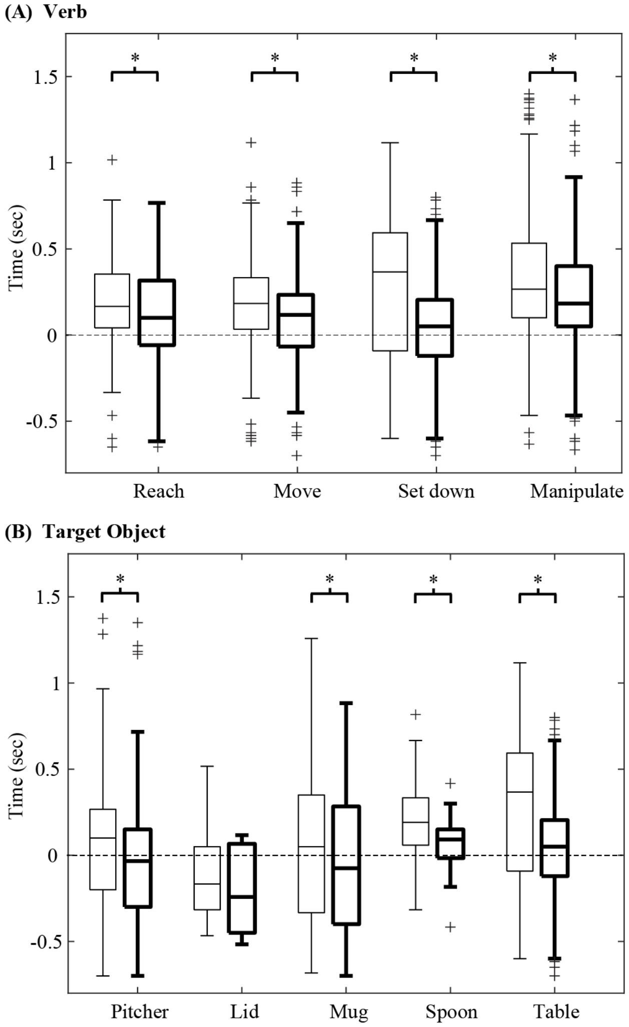


Figure 3.7: For a representative trial of Activity 1, temporal sequences of recognition results and ground truth are presented for (A) verb and (B) target object. In both (A,B), the top, middle, and bottom color bars represent the raw RNN output, RNN output smoothed by a mode filter, and hand-labeled ground truth, respectively. The total duration of this trial is 36 s.



(A) Verb

(B) Target Object

Figure 3.8: For Activity 1, the observational latency for recognition of (A) verb and (B) target object are shown using box and whisker plots. A negative latency value indicates that a verb or target object is identified before the start of the action primitive. For each boxplot pair, the observational latency without using GOA and GOAS (thin lines) is compared with that using GOA and GOAS (thick lines). Each boxplot indicates the 25, 50, and 75th percentiles. The whiskers extend to the most extreme data points that are not considered outliers ("+") having values of more than 1.5 times the interquartile range from the top or bottom of the box. Asterisks indicate $p < \alpha = 0.05$.

We aim to exploit observations that gaze behavior is a critical component of sighted grasp and manipulation activities, and that eye movements precede hand movements ([32, 106]). As such, we adopted the gaze-based feature GO from the literature (e.g., [17]) and supplemented it with two new features that we defined: GOA and GOAS.

As reported in section Effect of Input Features on Recognition Accuracy, models that included GOA and GOAS as input features outperformed models that relied primarily on GO or HO for every verb and target object class. The addition of GOA and GOAS substantially improved the average F1-score from 64% to 77% for verb and from 71 to 83% for target object (Figure 6).

The advantages of using features based on gaze object angle for action primitive recognition are 2-fold. First, the gaze object angle quantifies the spatiotemporal relationship between the gaze vector and every object in the workspace, including objects upon which the subject is not currently gazing. In contrast, the gaze object only captures the identity of the object upon which the subject is gazing at that particular instant. Considering that daily activities generally involve a variety of objects, it is vital for the classifier to collect sufficient information related to gaze-object interactions. The feature GOA could indirectly provide information similar to that of GO. For example, a GOA value that is close to zero would result if the gaze vector is essentially pointing at the gaze object. When GOA, GOAS, and HO have already been included as input features, the addition of GO as an input feature has little to no impact on classification accuracy (Figure 6). Also, classifier performance improves when using GOA and GOAS as input features as compared to using GO, HO, or their combination (Figure 6).

Second, the input feature GOAS contains GOA rate information. To some extent, GOAS also captures directional information, as positive and negative GOAS values reflect whether the gaze vector is approaching or departing from each object in the workspace, respectively. We believe that approach/departure information can be leveraged to predict the target object for a given action primitive because gaze is used to gather visual information for planning before and during manual activities ([106]). An object being approached by the gaze vector is not necessarily the target object, as the object could simply be in the path of the gaze vector during its movement. However, objects are less likely to be labeled as the "target object" when the gaze vector moves away from them.

3.6.2 Features Based on Gaze Object Angle Improve Observational Latency

While recognition accuracy is important, human-robot systems also require low observational latency ([75]). Even an action primitive that is correctly recognized 100% of the time will cease to be useful if the delay in recognition prohibits an effective response or adds to the cognitive burden of the operator. The earlier that a robotic system can infer the intent of the human operator or collaborator, the more time will be available for computation and the planning of appropriate robot movements.

Previous studies have focused on classifying actions in videos that have already been segmented in time (e.g., [19]. However, these methods that were designed to recognize actions in hindsight would be less effective for real-time use. We desire the intended action primitive to be predicted in advance of robot movement and with as low an observational latency as possible.

Hoffman proposed several metrics to evaluate fluency in human-robot collaborative tasks. For instance, the robot's functional delay was defined as the amount of time that the human spent waiting for the robot ([107]). This concept of fluency reflects how promptly a robot can respond correctly to an operator's commands. A high observational latency will degrade the fluency of a human-robot system and increase the operator's cognitive burden, effort, and frustration levels. A user interface that requires operators to intentionally gaze at specific objects or regions for a fixed period of time may be less natural and have lower fluency than a user interface that leverages natural eye gaze behaviors ([28, 90]).

In this work, the use of gaze-related features enabled the recognition of action primitives at an early stage. The average observational latency for verb recognition was 120 ms, 10% of the average duration of an action primitive (1.2 s). The average observational latency for target object was -50 ms; the negative latency value indicates that the target object was sometimes identified before the start of the action primitive. Unfortunately, pooled across all classes, the observational latency for the target object was not statistically significantly less than zero (p = 0.075; $\alpha = 0.05$). Nonetheless, the fact that some of the trials resulted in negative observational latency values was surprising and encouraging.

Among gaze-related input features, the use of GOA and GOAS decreased the observational latency as compared with using GO alone (Figure 8). Per a Wilcoxon signed rank test, observational latency was statistically significantly smaller when GOA and GOAS were used as input features than when they were excluded ($p < \alpha = 0.05$). This was true for all verb classes and all target object classes, with the exception of lid. For the verb and target object, the observational latency dropped by an average of 108 and 112 ms, respectively. One reason for this could be that GOA-based features may encode the tendency of the gaze vector to approach an object once the eyes start to move. In contrast, the GO feature does not capture the identity of any object until the gaze vector reaches the object.

The sub-second observational latency values that we report likely resulted from the fact that eye movement generally precedes hand movement for manual activities ([32, 106]). Land et al. reported that the gaze vector typically reached the next target object before any visible signs of hand movement during the activity of making tea ([61]). The small observational latency values may also result from the fact that our classifier was designed to recognize action primitives, which are much simpler than actions or activities ([93]). Action primitives often involve a single object, a single hand, and occur over a shorter period of time than actions and activities. The recognition of actions and activities for ADLs would require observations over a longer period of time and would necessarily involve more complex eye behaviors, more complex body movements, and gaze interactions with multiple objects.

Ryoo predicted activities of daily living and defined the "observation ratio" as the ratio between the observational latency and the activity duration ([108]). Ryoo reported that a minimum observation ratio of 45% was needed to classify activities with at least 60% accuracy. In this work, we found that minimum observation ratios of 18 and 5% were needed to achieve an accuracy of 60% for each the verb and the target object, respectively. This suggests that recognition of low-level action primitives can be achieved at lower observation ratios and within shorter time periods than high-level activities, which require the passage of more time and collection of more information for similar levels of accuracy.

One limitation of this work is that the action primitive recognition algorithm has not yet been tested in real-time. This is an area of future work and considerations for real-time implementation are discussed in section Comparisons to State-of-the-Art Recognition Algorithms. Based on our experience, we expect that the overall latency will be dominated by observational latency and less affected by computational latency. This is due to the relatively simple structure of the proposed RNN architecture and the fact that the RNN model would be trained offline a priori.

3.6.3 Segmenting Objects Into Regions According to Affordance Could Improve Recognition Performance

The distribution of gaze fixations can be concentrated on certain regions of an object, such as those associated with "object affordances." An object affordance describes actions that could be performed on an object ([57]). For example, Belardinelli et al. showed human subjects a 2D image of a teapot and instructed them to consider lifting, opening, or classifying the teapot as an object that could or could not hold fluid ([40]). It was observed that subjects' gaze fixations were focused on the teapot handle, lid, and spout for lifting, opening, and classifying, respectively. In addition, in a prior study, we reported 3D gaze heat maps for the activity "make a powdered drink" ([30]). We observed that gaze fixations were focused on the top and bottom of pitcher during the action unit "reach for pitcher" and "set down pitcher."

Inspired by these findings, we hypothesized that information about the action primitive can, in theory, be encoded by gaze behavior with respect to specific regions of objects. This would provide a classification algorithm with information at a finer spatial resolution than when considering each object as a whole. In a post hoc study, we segmented the point clouds of each of the four activity-relevant objects in Activity 1 (make a powdered drink) into several regions according to object affordances (Figure 9). For instance, the spoon was segmented into the upper and bottom faces for the bowl, the handle, and the tip of the handle. Notably, the inner and outer wall of containers (pitcher and mug) were treated as different regions since the inner and outer walls were often fixated upon differently depending on the action primitive.

After the segmentation, we augmented the gaze-related features (GO, GOA, GOAS) by treating each region as an independent object while keeping the features left-hand object and right-hand object unchanged. We then retrained the RNN with the new augmented features. The recognition accuracy for verb increased slightly from 77 to 79% and accuracy for the target object increased from 83 to 86%. By increasing the total number of object regions from 4 to 20, the time taken for the trained RNN to produce one classifier output increased by 26%. Depending on the consequences of an incorrect classification and the minimum acceptable accuracy level, one could decide which objects to segment and how finely the objects should be segmented. For instance, one may still be able to improve recognition performance if the mug were segmented into inner wall, outer wall, and handle, as opposed to the five segments that we tested.

3.6.4 Comparison to State-of-the-Art Recognition Algorithms

In the evaluation of our proposed gaze-based action primitive recognition method, we were unable to identify suitable benchmarks for a direct quantitative comparison. First, our approach is designed to recognize low-level action primitives that could be used as modular, generalizable building blocks for more complex levels of the action hierarchy ([93]). The literature on action recognition provides methods for recognition at the level of actions and activities, but not at the level of action primitives that are investigated in our work. For instance, the public dataset “GTEA+” and “EGTEA Gaze+” provided by [19, 109] involve actions such as “take bread.” This action would need to be split into two separate action primitives: “reach bread,” and “set down bread onto table.” Likewise, the public dataset “CMU-MMAC” provided by [44] involves actions such as “stir egg.” This action would need to be split into three action primitives: “reach fork,” “move fork into bowl,” and “stir egg in the bowl using fork.” Many state-of-the-art recognition methods for ADLs (whether

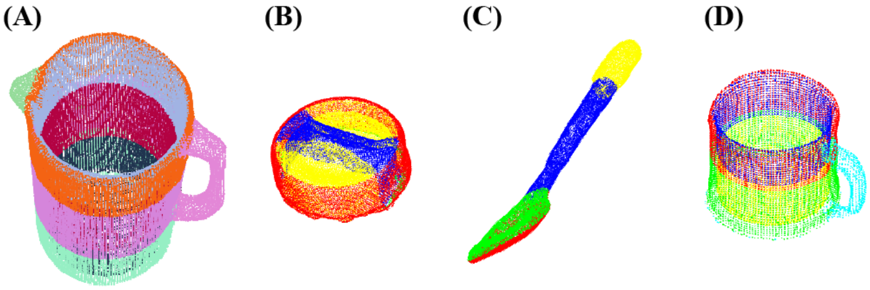


Figure 3.9: Point clouds of the four activity-relevant objects involved in Activity 1 were segmented into multiple regions for finer spatial resolution: (A) pitcher, (B) pitcher lid, (C) spoon, and (D) mug.

leveraging gaze behavior or not) are based on these publicly available datasets at the action level.

Second, action recognition models in the literature rely on computer-vision based approaches to analyze 2D videos recorded by an egocentric camera, e.g., ([19, 20, 22, 24, 23, 109, 110, 111, 112]). Whether using hand-crafted features ([18, 19, 20, 22, 24, 23, 110]) or learning end-to-end models ([109, 111, 112]), the computer vision-based approaches to action recognition must also address the challenges of identifying and tracking activity-relevant objects. In contrast, we bypassed the challenges inherent in 2D image analysis by combining an eyetracker with a marker-based motion capture system. This experimental set-up enabled the direct collection of 3D gaze-based features and object identity and pose information so that we could focus on the utility of 3D gaze features, which are unattainable from 2D camera images. Our method could be introduced into non-lab environments by combining an eyetracker with 2D cameras and ArUco markers, for example, in place of a marker-based motion capture system.

3.6.5 Considerations for Real-Time Implementation of an Action Primitive Recognition Algorithm in Human-Robot Systems

As an example of how our action primitive recognition model could be applied in a human-robot shared autonomy scenario, consider the action "stir contents inside a mug." First, as a subject's eye gaze vector moves toward the spoon, the probability of the potential action primitive "reach spoon" increases until it exceeds a custom threshold. The crossing of the threshold triggers the robotic end effector to move autonomously toward the spoon handle in order to grasp the spoon. The robot would use its real-time 3D model of the scene to plan its low-level movements in order to reduce the cognitive burden on the human operator. Second, as the subject's eye gaze switches to the mug after a successful grasp of the spoon, the model would recognize the highest probability action primitive as "move spoon to mug." Again the crossing of a probability threshold, or confidence level, would trigger the autonomous placement of the grasped spoon within the mug for a subsequent, allowable manipulate-type action primitive, which would be limited to a set of allowable manipulate-type action primitives based on the gaze object and hand object. Third, as the subject fixates their gaze on the mug, the model would recognize the highest probability action primitive as "stir inside mug" and autonomous stirring would begin. The stirring trajectory could be generated using parametric dynamic motion primitives [113], for example. Lastly, as the subject's gaze saccades to a support surface and the action primitive is recognized as "set down spoon," the system would proceed to determine a location on the table at which to place the spoon. This exact location could be extracted from filtered eye gaze signals as introduced in [14].

As described in the above example, we envision that our model could be used to recognize subjects' intended action primitives through their natural eye gaze movements while the robot handles the planning and control details necessary for implementation. In contrast to some state-of-the-art approaches to commanding robot movements [36, 90, 92, 91], subjects would not be forced to unnaturally, intentionally fixate their gaze at target objects in order to trigger pre-programmed actions. Of course, much work is necessary to implement the proposed shared autonomy control scheme and this is the subject of future work.

Concerning the practical implementation of the proposed action primitive recognition method, several limitations must be addressed.

3.6.6 Specificity of the Action Primitive

The proposed recognition method is intended to assign generalized labels to each time step as one of the four verb classes (reach, move, set down, and manipulate). The current method does not distinguish between subclasses of manipulate-type verbs, such as "pour" and "stir." Recognition of subclasses of a verb could enable assistive robots to provide even more specific assistance than that demonstrated in this work.

Recognition specificity could be advanced by incorporating additional steps. One idea is to create a lookup table based on the affordances of the objects involved in the activities. For example, the action primitive triplet of (verb = manipulate, TO = mug, HO = pitcher) is associated with the verb subclass "pour." However, the triplet (verb = manipulate, TO = pitcher, HO = spoon) is associated with both verb subclasses "stir" and "scoop." As an alternative, we suggest the use of gaze heat maps to facilitate the classification of verb subclasses since action primitives are activity-driven and the distribution of gaze fixations can be considerably affected by object affordance ([40, 30]).

3.6.7 Distracted or Idle Eye Gaze States

The proposed recognition method does not recognize human subjects' distracted or idle states. For example, a subject's visual attention can be distracted by environmental stimuli. In this study, we minimized visual distractions through the use of black curtains and by limiting the objects in the workspace to those required for the instructed activity. The incorporation of distractions (audio, visual, cognitive, etc.) is beyond the scope of this work, but would need to be addressed before transitioning the proposed recognition method to natural, unstructured environments.

Idle states are not currently addressed in this work. Hands are not used for every activity and subjects may also wish to rest. If the gaze vector of a daydreaming or resting subject happens to intersect with an activity-relevant object, an assistive robot may incorrectly recognize an unintended action primitive and perform unintended movements. This is similar to the "Midas touch" problem in the field of human-computer interaction, which faces a similar challenge of "how to differentiate 'attentive' saccades with intended goal of communication from the lower level eye movements that are just random" ([114]). This problem can be addressed by incorporating additional human input mechanisms, such as a joystick, which can be programmed to reflect the operator's agreement or disagreement with the robot's movements. The inclusion of "distracted" and "idle" verb classes would be an interesting area for future advancement.

3.6.8 Integration With Active Perception Approaches

The proposed recognition method could be combined with active perception approaches that could benefit a closed-loop human-robot system that leverages the active gaze of both humans and robots. In this work, the 3rd person cameras comprising the motion capture system passively observed the scene. However, by leveraging the concept of “joint attention” [115], one could use an external and/or robot-mounted camera set-up to actively explore a scene and track objects of interest, which could be used to improve the control of a robot in a human-robot system.

As discussed in section Comparisons to State-of-the-Art Recognition Algorithms, for the purposes of this work, we bypassed the process of identifying and locating activity-relevant objects by implementing a marker-based motion capture system in our experiment. Nonetheless, the perception of activity-relevant objects in non-laboratory environments remains a challenge due to object occlusions and limited field of view. Active perception-based approaches could be leveraged in such situations. In multi-object settings, such as a kitchen table cluttered with numerous objects, physical camera configurations could be actively controlled to change 3rd person perspectives and more accurately identify objects and estimate their poses [116]. Once multiple objects’ poses are determined, a camera’s viewpoint could then be guided by a human subject’s gaze vector to reflect the subject’s localized visual attention. Since humans tend to align visual targets with the centers of their visual fields ([117]), one could use natural human gaze behaviors to control camera perspectives (external or robot-mounted) in order to keep a target object, such as one recognized by our proposed recognition method, in the center of the image plane for more stable computer vision-based analysis and robotic intervention (Li et al., 2015a). When realized by a visible robot-mounted camera, the resulting bio-inspired centering of a target object may also serve as an implicit communication channel that provides feedback to a human collaborator. Going further, the camera’s perspective could be controlled actively and autonomously to focus on the affordances of a target object after a verb-TO pair is identified using our proposed recognition method. Rather than changing the physical configuration of a camera to center an affordance in the image plane, one could instead focus a robot's attention on an affordance at the image processing stage ([118]). For instance, the camera's foveal vision could be moved to a pitcher's handle in order to guide a robot's reach-to-grasp movement. Such focused robot attention, whether via physical changes in camera configuration or via digital image processing methods, could be an effective way to maximize limited computational resources. The resulting enhanced autonomy of the robot could help to reduce the cognitive burden on the human in a shared autonomy system.

Considering the goal of our work to infer human intent and advance action recognition for shared autonomy control schemes, one could also integrate our proposed methods with the concept of "active event recognition," which uses active camera configurations to simultaneously explore a scene and infer human intent [119]. Ognibene and Demiris developed a simulated humanoid robot that actively controlled its gaze to identify human intent while observing a human executing a goal-oriented reaching action. Using an optimization-based camera control policy, the robot adjusted its gaze in order to minimize the expected uncertainty over numerous prospective target objects. It was observed that the resulting robot gaze gradually transitioned from the human subject's hand to the true target object before the subject's hand reached the object. As future work, it would be interesting to investigate whether and how the integration of 1st person human gaze information, such as that collected from an ego-centric camera, could enhance the control of robot gaze for action recognition. For instance, the outputs of our proposed action primitive recognition method (verb-TO pairs) could be used as additional inputs to an active event recognition scheme in order to improve recognition accuracy and reduce observational latency.

3.6.9 Effects of the Actor on Eye Gaze Behavior

The proposed recognition model was trained using data in which non-disabled subjects were performing activities with their own hands instead of subjects with upper-limb impairment who were observing a robot that was performing activities. In our envisioned human-robot system, we seek to identify operator intent via their natural gaze behaviors before any robotic movements occur. It is known that gaze behaviors precede and guide hand motions during natural hand-eye coordination ([85]). In contrast, we hypothesize that the eye gaze behaviors of subjects observing robots may be reactive in nature. Aronsen et al. have shown that subjects' gaze behaviors are different in human-only manipulation tasks and human-robot shared manipulation tasks ([120]). The further investigation of the effect of a robot on human eye gaze is warranted, but is beyond the scope of this work. We propose that the eye gaze behaviors reported in this work could be used as a benchmark for future studies of human-robot systems that seek to recreate the seamlessness of human behaviors.

The direct translation of the model to a human-robot system may not be possible. For one, the robot itself would need to be considered as an object in the shared workspace, as it is likely to receive some of the operator's visual attention. Fortunately, as suggested by Dragan and Srinivasa in [121], the action primitive prediction does not need to be perfect since the recognition model can be implemented with a human in the loop. The robotic system could be designed to wait until a specific confidence level for its prediction of human intent has been achieved before moving.

Another important consideration is that the recognition of action primitives via human eye gaze will necessarily be affected by how the robot is programmed to perform activities. For example, eye gaze behaviors will depend on experimental variables such as manual teleoperation vs. preprogrammed movements, lag in the robot control system and processing for semi-autonomous behaviors (e.g., object recognition), etc. Recognizing that there are innumerable ways in which shared autonomy could be implemented in a human-robot system, we purposely elected to eliminate the confounding factor of robot control from this foundational work on human eye-hand coordination.

3.6.10 Integration of Low-Level Action Primitive Recognition Models With Higher Level Recognition Models

This work focused on the recognition of low-level action primitives. However, the envisioned application to assistive robots in a shared autonomy schema would require recognition at all three hierarchical levels of human behavior (action primitives, actions, activities) [93] in order to customize the degree of autonomy to the operator [122, 27]. For instance, the outputs of the low-level action primitive recognition models (such as in this work) could be used as input features for the mid-level action recognition models (e.g., [30], that would then feed into the high-level activity recognition models ([17]). Simultaneously, knowledge of the activity or action can be leveraged to predict lower level actions or action primitives, respectively.

3.7 Conclusion

The long-term objective of this work is to advance shared autonomy by developing a user-interface that can recognize operator intent during activities of daily living via natural eye movements. To this end, we introduced a classifier structure for recognizing low-level action primitives that incorporates novel gaze-related features. We defined an action primitive as a triplet comprised of a verb, target object, and hand object. Using a non-specific approach to classifying and indexing objects, we observed a modest level of generalizability of the action primitive classifier across activities, including those for which the classifier was not trained. We found that the gaze object angle and its rate of change were especially useful for accurately recognizing action primitives and reducing the observational latency of the classifier. In summary, we provide a gaze-based approach for recognizing action primitives that can be used to infer the intent of a human operator for intuitive control of a robotic system. The method can be further advanced by combining classifiers across multiple levels of the action hierarchy (action primitives, actions, activities) [93] and finessing the approach for real-time use. We highlighted the application of assistive robots to motivate and design this study. However, our methods could be applied to other human-robot applications, such as collaborative manufacturing.

CHAPTER 4

Gaze-based Shared Autonomy Framework with Real-time Action Primitive Recognition for Robot Manipulators

4.1 Abstract

Robots capable of robust, real-time recognition of human intent during manipulation tasks could be used to enhance human-robot collaboration for activities of daily living. Eye gaze-based control interfaces offer a non-invasive way to infer intent and reduce the cognitive burden on operators of complex robots. Eye gaze is traditionally used for "gaze triggering" (GT) in which staring at an object, or sequence of objects, triggers pre-programmed robotic movements. We propose an alternative approach: a neural network-based "action prediction" (AP) mode that extracts gaze-related features to recognize, and often predict, an operator's intended action primitives. We integrated the AP mode into a shared autonomy framework capable of 3D gaze reconstruction, real-time intent inference, object localization, obstacle avoidance, and dynamic trajectory planning. Using this framework, we conducted a user study to directly compare the performance of the GT and AP modes using traditional subjective performance metrics, such as Likert scales, as well as novel objective performance metrics, such as the delay of recognition. Statistical analyses suggested that the AP mode

resulted in more seamless robotic movement than the state-of-the-art GT mode, and that participants generally preferred the AP mode.

4.2 Introduction

Activities of daily living (ADLs) can be challenging for individuals with upper limb impairment. Assistive robotic arms can significantly increase one's functional independence by easing the performance of ADLs [123]. However, the direct control of robotic arms with numerous degrees-of-freedom (DOFs) via low dimensional input devices, such as a joysticks, imposes a high cognitive burden on operators. Operators must frequently switch between several modes for commanding gripper position, orientation, and open/close, and do so using an unintuitive 3D Cartesian space perspective. To make the control process more intuitive and seamless, we pursued a "shared autonomy" approach in which operator inputs and semi-autonomous control are integrated in order to achieve shared goals [15].

A variety of non-verbal human input interfaces have been leveraged to recognize human intent, such as whole-body interfaces [124], gestures [77, 83, 84], electromyography (EMG) [78], electroencephalography (EEG) [79, 80], and electrocorticography (ECoG) [76, 82]. In this work, we use an eye tracker due to advantages, such as being non-invasive, non-verbal, intuitive, and easy to don and doff.

In prior studies, eye gaze was simply used as a "cursor" to select a target object from several candidate objects [125, 91, 126, 90, 92, 127]. These conventional methods did not attempt to infer or predict intent and required operators to stare at a target object for fixed duration in order to trigger a pre-programmed robotic trajectory. We refer to such a control approach as the "gaze trigger" (GT) method. In addition, prior studies mainly focused on

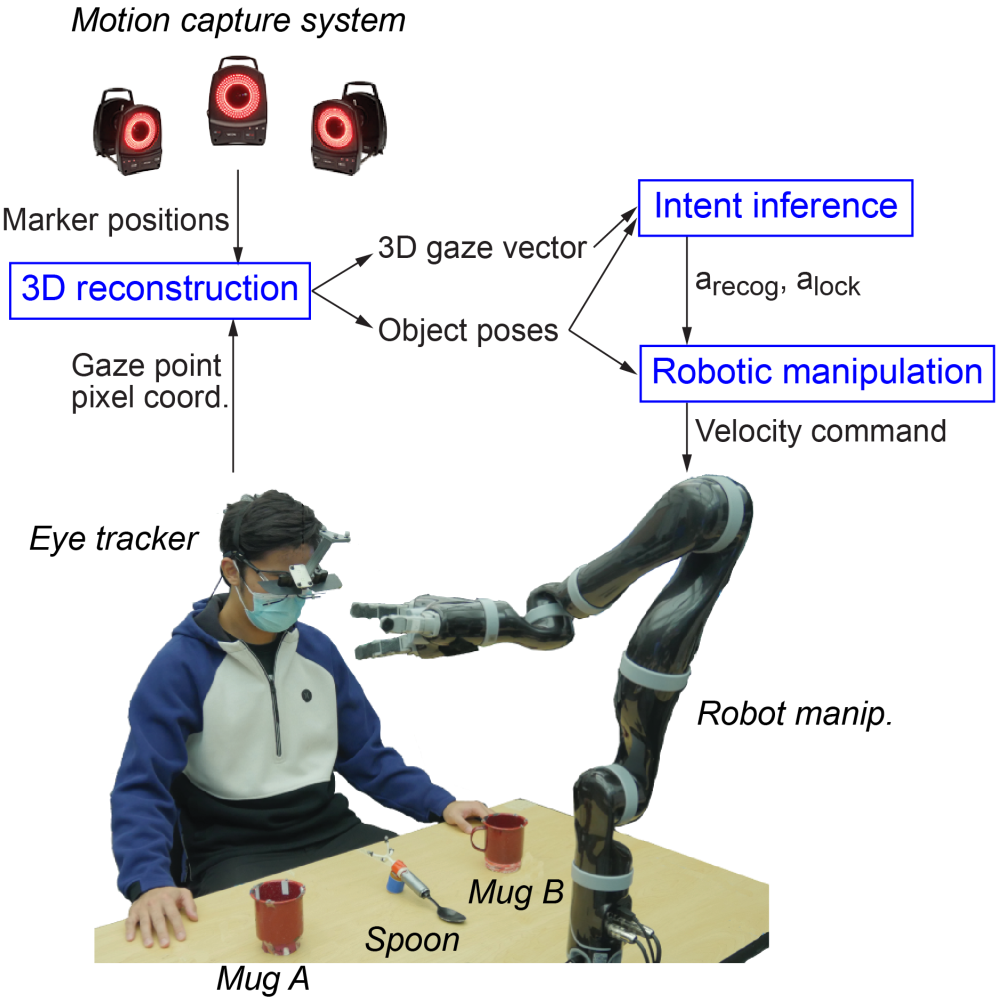


Figure 4.1: The gaze-based shared autonomy framework consists of three threads: 3D reconstruction, intent inference, and robotic manipulation. Gaze-related features are used to recognize action primitives to enable seamless robotic movements during the assistance of activities of daily living.

pick-and-place tasks [125, 126, 90, 127].

The objective of this work is to enhance gaze-based shared autonomy systems by introducing a neural network-based “action prediction” (AP) algorithm that leverages spatiotemporal gaze-related features. We propose a number of objective and subjective performance metrics to evaluate and compare the performance of two control modes: the state-of-the-art GT mode and our proposed AP mode. There are two main contributions of this study. First, we developed and implemented an “action prediction” control mode for a gaze-based shared autonomy framework that can be used to perform everyday tasks comprised of a sequence of actions. The system features capabilities for intent inference, object localization, obstacle avoidance, and dynamic trajectory planning. Second, we demonstrated that the AP control model results in more seamless robotic movements than the state-of-the-art GT mode, and that participants often preferred the proposed AP control mode over the GT mode.

This article is organized as follows. Section 4.3 outlines related work concerning gaze-based action recognition and gaze-based shared autonomy. Section 4.4 introduces our proposed gaze-based shared autonomy framework and action prediction control mode, and Section 4.5 describes the experimental evaluation of the framework. Section 4.6 presents a comparison of the performance of the state-of-the-art gaze trigger control mode and the proposed action prediction control mode. Section 4.7 concludes with a summary of contributions.

4.3 Related Work

4.3.1 Gaze-based Action Recognition

Numerous computer vision-based studies have leveraged egocentric videos taken by head-mounted cameras or eye trackers to recognize actions during everyday tasks [16, 17, 18, 19, 20, 21, 22, 23, 24]. These studies first subtracted the foreground and then detected human hands and activity-relevant objects. Features related to hands, objects, gaze, and their relative spatial relations were then used as inputs for action recognition using approaches such as HMMs, neural networks, and support vector machines (SVMs). Actions could not be successfully recognized until key visual features related to hand motions and object states (e.g. whether the lid is on a cup) were available to the classification algorithm. In this work, we aim to *predict* an operator's intended actions using gaze-related features. Thus, computer vision-based action recognition algorithms that rely on the visual consequences of actions cannot be directly applied for intent inference prior to the initiation of actions.

Li and Zhang proposed a gaze-based intention communication framework for human-robot interaction that was designed for eventual use with an assistive robot [28]. A simulated kitchen image was displayed to subjects who were instructed to express their intent by looking at task-relevant objects in the image. Subjects were required to press a physical button before and after they expressed their intention using visual attention in order to identify the sequence of gazed objects to be used for SVM classification of intent. While the system enabled recognition of intended tasks, such as "prepare a cup of coffee," a number of steps were required of the operator, thereby reducing the intuitive nature of control and seamlessness of the shared autonomy system.

Fuchs and Belardinelli recorded gaze signals as operators used a gaming controller to control the 3D position of a virtual robot end-effector in order to perform a pick-and-place task [29]. The gaze point was fed into a Gaussian Hidden Markov Model to classify a verb ("pick" or "place") and target (cylinders to be grasped or locations for setting down grasped cylinders). Although a recognition accuracy of approximately 80% was achieved, the eye gaze signal was interpreted as a gaze point rather than a 3D gaze vector, and the action recognition was not tested with a real robot or for tasks other than pick-and-place.

In a gaze-based intent inference study conducted by Huang et al., a "customer" selected one ingredient at a time for the preparation of a sandwich by a "server" [128]. Using gaze-based features, an SVM-based method correctly predicted the selected ingredient approximately 1.8 sec before a verbal request was given. While intent inference was successfully implemented for a target object (ingredient), the study did not incorporate the prediction of any verbs, as it was assumed that each ingredient was to be added to the sandwich.

In a prior study, we interpreted human intent as a triplet of a verb, target object, and hand object [71]. In that study, we recruited subjects to perform several everyday activities, such as preparing a powdered drink, and trained a recurrent neural network (RNN) to simultaneously recognize verbs and target objects using gaze-based features. As detailed in Section 4.4.4, we leverage our prior RNN-based action recognition algorithm in this work.

4.3.2 Shared Autonomy Systems with Gaze-based Robot Control

While the works cited in Section 4.3.1 addressed the challenge of action recognition using eye gaze, the recognition algorithms were not implemented in shared autonomy systems with real robots. In this Section 4.3.2, we provide an overview of works that implemented the gaze-based control of real robots.

Previous studies on gaze-based shared autonomy have focused on pick-and-place tasks [90, 127]. Gaze was used to select a target object for pick-up or a target position for setting down a grasped object. A robotic action would be triggered once gaze fixation on a target exceeded a preset time threshold (e.g. 2 sec in [90]). Zeng, et al. used a hybrid gaze-brain machine interface in order to trigger robotic actions [125, 91, 126]. Gaze was used to select target objects while an EEG brain-machine interface triggered an action using “motor imagery” data. In the aforementioned studies, gaze was used to identify objects for pre-programmed movements.

Shafti et al. expanded the repertoire of gaze-triggered actions by adding pouring to pick and place capabilities [92]. A finite state machine was used to select the next action based on the identities and affordances [57] of the grasped and gazed objects. For instance, when the grasped object is a mug and the gazed object is a bowl, the next action to be triggered is “pour.” In this work, we build upon the finite state machine concept, but we incorporate an action recognition algorithm and create a more generalizable decision-making structure that is based on object affordances instead of specific object identities.

Admoni and Srinivasa proposed an intent inference method for gaze-based shared autonomy systems [25]. A Partially Observable Markov Decision Process model used joystick and eye tracker signals in order to update probability distributions for candidate actions. However, we were unable to find follow-on publications in which the proposed algorithm was demonstrated experimentally.

Huang and Mutlu demonstrated a gaze-based intent inference method for human-robot interaction [26]. A “customer” ordered a drink by verbally requesting one ingredient at a time while a robotic “server” picked up the corresponding ingredient and placed it into a blender. The robotic system monitored the customer’s gaze, predicted the intended ingredient using

SVM-based classification, and acted proactively. With the intent inference algorithm and the proactive control method, the system could respond to a customer's request and complete the task 2.5 sec earlier on average. In this work, we aim to predict a verb in addition to a target object, and to develop a larger repertoire of verbs and robotic actions.

4.4 Gaze-based Shared Autonomy Framework

Our proposed gaze-based shared autonomy framework consists of three threads: 1) 3D reconstruction, 2) intent inference, and 3) robotic manipulation (Figure 4.1). The 3D reconstruction thread tracks the 3D gaze vector as well as the location and orientation of task-relevant objects. The intent inference thread extracts input features from 3D gaze-object spatiotemporal data and feeds the features into a recurrent neural network (RNN) in order to perform real-time recognition of the intended action primitive. The robotic manipulation thread executes the intended action primitive while also implementing collision avoidance. In addition, a high-level control logic integrates the three parallel threads and enables the sequential execution of one action primitive after another. This section presents how each thread was designed and integrated into a system.

4.4.1 Intent Representation

Before we introduce the control logic and three parallel threads in the gaze-based shared autonomy framework, we first define operator intent. Leveraging our prior work [71], we represent operator intent as an action primitive triplet comprised of a verb, target object (TO), and hand object (HO).

The verb can be one of four classes: Reach, Set down, Manipulate, or Move. The verb class Manipulate includes a list of manipulate-type verbs that are highly related to object-specific affordances [57]. For instance, the verb "stir" is closely associated with the object spoon, and the verb "pour" is closely associated with the object mug.

The target object (TO) refers to the object or support surface that will be directly affected by verbs. The hand object (HO) refers to the object grasped by the robotic gripper. For instance, in the action primitive "move the spoon to the mug", the verb, HO, and TO are "move," "spoon", and "mug," respectively.

4.4.2 Control Logic

A high-level control logic in Procedure 4.1 integrates the three parallel threads of the shared autonomy framework (3D reconstruction, intent inference, robotic manipulation). The parallel threads each operate at 100 Hz. First, the 3D reconstruction thread is initiated so that the robot always has access to the real-time 3D gaze vector as well as the locations and orientations of the task-relevant objects. The system then enters a `while` loop to recognize and execute action primitives until all of the action primitives necessary for the task have been completed.

To improve the robustness of the robot actions taken based on RNN classification, we used a finite state diagram that efficiently prunes action primitive candidates based on common sense. A state is comprised of an "HO flag" and a "manipulate flag." For instance, when no object is grasped by the robotic gripper, the HO flag is set to "NA" ("not applicable"), and only action primitives with the verb "reach" are considered as viable and treated as action primitive candidates. Immediately after an object, such as a spoon, has been grasped, the HO flag is updated to "≠NA" and the manipulate flag is set to "not ready to manipulate". The robot will not be allowed to execute manipulate-type action primitives, such as "stir," until it becomes logical to do so, such as when the spoon is moved into a mug ("HO≠NA," manipulate="ready to manipulate").

During a single iteration of the `while` loop, the system updates the "ready to manipulate" flag according to the spatial relationship between the target object and hand object, and then updates the list of action primitive candidates that will be used by the intent inference thread. The system plans the target pose or trajectory associated with each action primitive candidate as proposed in Sections 4.4.5.1 and 4.4.5.2, respectively. After the intent inference and robotic manipulation threads are started, the robotic manipulation thread retains control until an action primitive is completed (i.e. until an "action-complete" flag switches from false to true). After an action primitive is completed, the intent inference and robotic manipulation threads are stopped, and the "action-complete" flag is reset to false such that the robot is prepared for subsequent actions.

4.4.3 3D Reconstruction of Gaze Vector and Objects

Using a motion capture system and eye tracker described in Section 4.5, the participant's 3D gaze vector was reconstructed. The 3D gaze vector points from the origin of the egocentric camera frame to the gaze point location in the image plane. In addition to the gaze vector, each object's pose and point cloud were updated in real-time using markers [30].

4.4.4 Intent Inference

4.4.4.1 Action Primitive Recognition

In order to recognize participants' intended action primitives, we deployed a pre-trained RNN whose training process was detailed in our prior work [71]. The training dataset in

Procedure I: Control Logic

```
Start the 3D reconstruction thread
while task not completed do for each action primitive in the task
    Update “ready to manipulate”
    Update the set of action primitive candidates according to the finite state diagram
    Plan the target poses/trajectories for each action primitive candidate
    Start the intent inference and robotic manipulation threads
    while (“action-complete” is false) do
        Continue to the intent inference and robotic manipulation threads
    endwhile
    Stop the intent inference and robotic manipulation threads
    Set “action-complete” as false
endwhile
```

[71] was drawn from experiments in which participants performed three everyday tasks: making a powdered drink, making instant coffee, and preparing a cleaning sponge. The RNN anonymized the identities of the task-relevant objects via a generic sorting and indexing method in order to improve cross-task generalizability. As a result, the RNN is not dependent on object identity or the specific task.

Four types of features were extracted as inputs to the RNN: gaze object, hand object, gaze object angle, and gaze object angular speed. The gaze object is the first object to be intersected by the 3D gaze vector. While the hand object was defined in [71] as the object grasped by a participant’s dominant hand, the hand object in this work is defined as the object held by the robotic gripper. Gaze object angle is defined as the angle between the gaze vector and the eye-object vector. The eye-object vector emanates from the origin of the gaze vector, but ends at an object's center of mass. Finally, gaze object angular speed is the time derivative of the gaze object angle. The output of the RNN is a probability distribution among action primitive candidates.

4.4.4.2 Implementation of the Intent Inference Thread

The intent inference thread (Procedure 4.2) collects gaze-related features and sends them as inputs to the RNN. The real-time, action primitive recognition RNN outputs are probability values whose noise is reduced through the use of a moving average filter window $W_{f,recog}$ of length $w_{f,recog}$. The filter is applied to the probability values $P_{a_c}(i)$ for each of $c = 1, \cdots, n_{cand}$ action primitive candidates a_c for all time steps i within the filter window. The filter output is an average, normalized probability $\overline{P}_{a_c}$ for each action primitive candidate. The normalized probability values for the action primitive candidates sum to one.

The real-time action primitive recognition output a_{recog} is set as the action primitive candidate with the highest average, normalized probability.

$$a_{recog}(i) = \underset{c}{\operatorname{argmax}}(\overline{P}_{a_c}(i \in W_{f,recog})), \quad c = 1, \cdots, n_{cand} \tag{4.1}$$

Despite the use of a moving average filter on the RNN output, a strict implementation of the real-time action primitive recognition output can still result in unsmooth robot behavior. Thus, we implemented a "locking" mechanism to enable the completion of a given action primitive. In order for an action primitive to be locked, two criteria must be simultaneously satisfied: (i) the Euclidean distance between the end-effector and the target location associated with a_{recog} must be less than a user-defined distance threshold d, and (ii) the average, normalized probability of the recognized action primitive $\overline{P}_{a_{recog}}(t \in W_{lock})$ must exceed a user-defined probability threshold p_{lock}. A user-defined "locking" window W_{lock} of length w_{lock} determines which timesteps to consider when calculating the average, normalized probability for a recognized action primitive.

When the action primitive a_{recog} is locked, the robot will ignore subsequent action primitive recognition outputs from the RNN until the execution of the action primitive has been completed or an "unlocking" criterion has been met. The action primitive a_{lock} can be unlocked when its average, normalized probability during the locking window $\overline{P}_{a_{lock}}(t \in W_{lock})$ falls below a user-defined probability threshold p_{unlock}. Once a locked action primitive is unlocked, the ongoing robotic manipulation thread is terminated.

4.4.5 Robotic Manipulation

In this section, we detail our methods for the planning and implementation of gaze-based action primitives on real robots. We address the planning of target poses and movement trajectories that are efficient and capable of avoiding obstacles.

4.4.5.1 Planning Target Poses

For the verb "reach," we defined a discrete set of target poses for each object according to the object's affordances. For example, a mug is most commonly grasped from the side by its body or handle. The target poses are fixed relative to the object and translate and rotate in 3D along with the object. Before initiating any reaching movement, the robot prunes target pose candidates that would result in collisions and selects the target pose that is closest to the current end-effector location.

Procedure II: Intent Inference Thread

```
Collect gaze-related features and send them to the RNN
Generate a normalized probability distribution for all action
    primitive candidates using the RNN
Update $a_{recog}$ using the moving average filter (eq. 4.1)
Update recognition history in $W_{f,recog}$ and $W_{lock}$
if no action primitive is locked then
    if locking criterion is satisfied then
        $a_{lock} = a_{recog}$
    endif
else an action primitive is already locked
    if ("action-complete" is true) or (unlock. criterion satisfied) then
        Unlock the locked action
        Set "action-complete" as true
    endif
endif
```

For the verb "move," target poses are planned for each action primitive candidate that might follow. For example, if a manipulate-type verb is to follow the verb "move," then the target pose for "move" is set as the end-effector's initial pose for the subsequent manipulate-type verb.

For the verb "set down," an operator's intended target position for setting down a grasped object on a support surface can be hidden within unfocused gaze signals, blinks, saccades, and involuntary eye movements. Unlike the verbs "reach" and "move," whose associated target positions can be determined by the target objects' locations and affordances, the intended target position on a support surface for the verb "set down" needs to be extracted from noisy eye gaze signals.

We adopted Li, et al.'s "fuzzy interpretation" method to filter out noise in eye gaze signals and extract valid points of visual attention [14]. Consider the point of intersection between the 3D gaze vector and support surface as a raw, unfiltered gaze point x_i. The variables x_i and $\widetilde{x}_i$ represent the i^{th} raw gaze point and the gaze point after being processed by the filter, respectively, at the time step i. We calculate the distance between the gaze point x_i and the geometric center of the cluster of "influential" gaze points in a moving filter window $W_{f,gaze}$ of length $w_{f,gaze}$. Per [14], if the distance is less than a user-defined threshold d_r, then the "influence coefficient" e_i is set equal to 1 and the gaze point x_i is added to the cluster of influential gaze points in $W_{f,gaze}$. Otherwise, the "influence coefficient" e_i is set equal to zero, and the gaze point x_i is discarded.

$$e_i = \begin{cases} 1, & \text{if } \left\| x_i - \dfrac{\sum\limits_{k=i-w_{f,gaze}}^{i-1} \widetilde{x}_k e_k}{\sum\limits_{k=i-w_{f,gaze}}^{i-1} e_k} \right\| < d_r \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Influential gaze points are used to calculate the average gaze point $\widetilde{x}_i$ within the moving filter window at the time step i (eq. 4.3). The moving filter window includes all time steps between the time steps $i - w_{f,gaze}$ and $i - 1$.

$$\widetilde{x}_i = \frac{\sum_{k=i-w_f}^{i-1} (\widetilde{x}_k e_k) + x_i e_i}{\sum_{k=i-w_f}^{i} e_k} \tag{4.3}$$

When at least 80% of the gaze points in $W_{f,gaze}$ are influential, we consider $\widetilde{x}_i$ as the participant's target position for setting down a grasped object.

4.4.5.2 Planning Movement Trajectories

For manipulate-type verbs, such as "pour" and "stir," we defined smooth trajectories that mimicked human demonstrations drawn from [71]. First, a time series of end-effector target poses are designed in Cartesian space such that the spatiotemporal relation between the target object and hand object remains the same as observed in human demonstrations. Following the use of inverse kinematics to convert the target poses from Cartesian space to joint angles, we used an iterative parabolic time parameterization method provided by MoveIt! to plan the joint velocities for execution on the robot [129].

4.4.5.3 Collision Avoidance via Artificial Potential Fields

For collision avoidance, we adopted a path planning framework based on the artificial potential field (APF) algorithm. The APF algorithm was introduced in 1985 by Khatib to quickly generate collision-free paths for robots in cluttered environments [130]. Attractive potential fields around goal locations would attract robot end-effectors while repulsive potential fields around obstacles would push end-effectors away.

Considering the irregular geometries of robot arms and end-effectors, grasped objects, and obstacles, we cannot simply represent each body as a particle, as is typically done for mobile robots. Leveraging the work of Khatib [130], we selected a set of "points subjected to potentials" ("PSPs") for each body. For the end-effector, we assigned a PSP to the tip of each digit in order to protect the non-backdrivable gripper from collision damage. Attractive and repulsive potential fields were generated based on the 3D positions of the PSPs.

One limitation of the original APF algorithm [130] is that interactions between attractive and repulsive potential fields may make some goals non-reachable. This problem, known as "goals non-reachable with obstacle nearby" (GNRON), refers to the undesirable situation when the goal location is not a minimum of the total potential field. An end-effector could get trapped in a local minimum near the goal, but never reach the goal. Thus, we adopted a modified repulsive potential function proposed by Zhu, et al. in 2006 that directly addresses the GNRON problem and enables the end-effector to reach its goal while also avoiding nearby obstacles [131].

4.4.5.4 Implementation of the Robotic Manipulation Thread

The robotic manipulation thread in Procedure 4.3 is used to execute action primitives on the robot hardware. When no action primitive is locked, the robotic manipulation thread sends a Cartesian velocity command to the robot according to the collision-free path planned using the APF algorithm.

When the locked action primitive verb is "reach," "move," or "set down," the robot ignores the real-time recognition result a_{recog} and prioritizes movement of the end-effector toward the target pose corresponding to a_{lock}. After the end-effector arrives at the target pose, depending on the verb of a_{lock}, it opens or closes the gripper, or sets the "ready to manipulate" flag to "true" in order to execute a manipulate-type verb.

When the locked action primitive verb is of the manipulate-type ("pour" or "stir"), the robot plans and executes a trajectory patterned after a human demonstration.

Procedure III: Robotic Manipulation Thread

```
if no action primitive is locked then
    Drive EEF toward the target pose associated with $a_{recog}$ using artificial potential fields
else an action primitive is already locked
    if verb of $a_{lock}$ is not Manipulate-type then
        if target pose not achieved then
            Drive EEF toward the target pose associated with $a_{lock}$ using artificial potential fields
        else target pose achieved
            case verb of $a_{lock}$ of
                Reach: Send close command to gripper, Update HO
                Set down: Send open command to gripper, Update HO
                Move: Set "ready to manipulate" as true
            endcase
        endif
    else verb of $a_{lock}$ is Manipulate-type
        Plan manipulate-type trajectory for $a_{lock}$
        Execute planned trajectory
    endif
endif
```

4.4.6 Control Modes

In a conventional gaze-based shared autonomy system, a robotic action is not triggered until gaze fixation on a target object exceeds a user-defined duration threshold [90, 92, 127].

We refer to this conventional control mode as the “gaze trigger (GT)” mode and use it as a benchmark for comparison with our proposed “action prediction (AP)” mode. Our intent inference model was integrated into the control scheme of the AP mode only.

We implemented the GT and AP modes under the same algorithmic framework having three parallel threads, as described in Section 4.4.2. However, there were three key differences in the practical implementation of the GT and AP modes due to the inclusion of the intent inference model in the AP mode. First, the prediction thread of the GT mode does not recognize intent using the RNN-based method of the AP mode (steps 1-3 in Procedure 4.2).

Second, when no action primitive is locked, the robotic manipulation thread of the GT mode does not send any velocity command to the robot while the AP mode does (steps 1-2 in Procedure 4.3).

Third, the locking and unlocking criteria are different. For the AP mode, the locking and unlocking criteria rely on the average probability of a_{recog} and the Euclidean distance between the end-effector and the target pose, as described in Section 4.4.4.2. For the GT mode, since the RNN-based method was not leveraged, the locking and unlocking criteria depend solely on gaze fixation.

The same locking window size W_{lock} was used for both the AP and GT modes. An action primitive is locked if gaze fixation on a target object exceeds 70% of the W_{lock} duration. An action primitive is unlocked if gaze fixation on a different target object exceeds 70% of the W_{lock} duration.

4.5 Experimental Evaluation

4.5.1 Experimental Protocol

We hypothesized that the action prediction (AP) mode would result in more seamless robotic movements than the state-of-the-art gaze trigger (GT) mode, and that participants would prefer the AP mode. In order to test these hypotheses, we conducted a study approved by the UCLA Institutional Review Board. All 16 participants (13 male, 3 female; aged 18-35 years) gave written informed consent in conformity with the Declaration of Helsinki. Three out of the 16 participants reported prior experience in interacting with robots.

We used a retro-reflective marker-based motion capture system (T-Series, Vicon, Culver City, CA, USA) with a sampling rate of 100 Hz to reconstruct the 3D gaze vector and to identify and locate task-relevant objects. An eye tracker (ETL-500, ISCAN, Inc., Woburn, MA, USA) having a sampling rate of 60 Hz provided 2D pixel coordinates that represented the perspective projection of an operator's gaze point onto the image plane of the eye tracker's egocentric camera. A traditional chessboard calibration procedure [100] and the MATLAB Camera Calibration Toolbox [47] were used to correct camera distortion and locate the origin of the egocentric camera frame with respect to markers attached to the eye tracker.

As shown in Figure 4.1, we used a 7 degree-of-freedom (DOF) robot arm (JACO^2 7DOF spherical, Kinova Robotics, Quebec, Canada) with a three-fingered end-effector (Kinova Robotics). For simplicity, we controlled the grip aperture of the end-effector only, effectively reducing the end-effector to a 1 DOF gripper. The experiment was conducted with a single computer having an Intel 9700K processor running at 3.6 GHz and an NVIDIA GeForce RTX 2070 GPU to accelerate the RNN calculations.

We selected everyday objects and actions common in activities of daily living for the assessment of the GT and AP modes within our gaze-based shared autonomy framework. We used three objects (mug A, mug B, spoon) from the benchmark Yale-CMU-Berkeley (YCB) Object set [43] and defined one support surface (table).

Each participant was instructed to control the robot to perform 10 actions in the following sequence: reach mug A, set down mug A, reach spoon, move spoon to mug A, stir within mug A, set down spoon, reach mug A, move mug A to mug B, pour from mug A into mug B, and set down mug A. Unlike most studies that focus solely on pick-and-place actions, we included actions that involve the verbs "move," "pour," and "stir." We indexed the sequentially performed action primitives as a_j, where $j \in \{1, \cdots, 10\}$. Collectively, these 10 actions involved a total of 14 unique action primitive candidates. For brevity and consistency across experimental sessions, we did not instruct participants to use action primitives in which mug B was the primary object of interest (e.g. stir within mug B).

For a consistent comparison of the GT and AP modes across subjects and trials, the sequence of 10 actions was prescribed through verbal instructions and objects were placed at preset locations before each new trial. However, there is nothing about the system implementation described in Section 4.4.2 that would prevent participants from improvising and changing the sequence of actions, or that relies upon prescribed locations for the task-relevant objects.

Each experimental session consisted of two blocks of trials, with each block consisting of one type of control mode and three consecutive repetitions/trials of that same control mode. To account for the possibility that the order of the blocks could bias results, half of the participants (selected at random) experienced the GT mode first while the remaining half experienced the AP mode first.

Each participant was instructed on how to control the robot for each mode with a script:

"You can let the robot know your intent by looking at the target object." Each participant was allowed to familiarize themselves with each control mode for up to two practice trials. Between each block, the participant was informed that the control mode would be switched. However, each control mode was referred to only as "Control Mode #1" or "Control Mode #2." As will be described in the following Section 4.5.2.3, participants were instructed to complete a brief questionnaire after each trial and were interviewed upon completion of the entire experimental session.

4.5.2 Performance Metrics

Here, we describe the objective and subjective performance metrics that were used to compare the performance of our proposed AP mode with the conventional GT mode.

4.5.2.1 Preliminaries

Before we define metrics for the seamlessness of the shared autonomy system, we introduce several key temporal variables. First, we consider an action primitive a_j, which is one of the instructed, sequentially performed action primitives where $j \in \{1, \cdots, 10\}$. The curves in Figure 4.2 represent the probability of a_j being the participant's intended action primitive. We define $t_{recog}(a_j)$ as the time at which a_j is *first* identified as a_{recog} according to eq. 4.1. We define $t_{end}(a_j)$ as the time at which a_j ends.

Importantly, our gaze-based shared autonomy framework allows for recognition of a_j prior to $t_{end}(a_{j-1})$ at which time the prior action primitive ends. However, any recognition of a_j earlier than a predefined time window W_{end} that immediately precedes $t_{end}(a_{j-1})$, i.e., the hatched area in Figure 4.2b, is treated as a possible misclassification and is ignored. The value of W_{end} determines the earliest time at which an action primitive may be predicted.

From an implementation perspective, it could be premature to take $t_{recog}(a_j)$ as the moment when the robot has correctly identified an intended action primitive, especially if the identity of a_{recog} changes from one time step to the next. Rapid changes in the identity of a_{recog} could occur due to noisy inputs to the RNN and despite the moving average filter applied to the RNN outputs. Thus, we conservatively define $t_{stable}(a_j)$ as the time of "stable" recognition. Occurring after $t_{recog}(a_j)$ for the AP mode, $t_{stable}(a_j)$ is the first time at which the following conditions are simultaneously satisfied: (i) at $t_{stable}(a_j)$, a_j is identified as a_{recog} according to eq. 4.1, and (ii) more than 70% of the time steps within a user-defined time window W_{stable} (solid grey area in Figure 4.2b) after $t_{stable}(a_j)$ are recognized as a_j. Note that for the GT mode, $t_{stable}(a_j)$ is the same as $t_{recog}(a_j)$ since both times correspond to the instant at which a_j is locked.

4.5.2.2 Objective Measures

We used the following objective measures to evaluate the seamlessness of the shared autonomy system: delay of recognition, delay of stable recognition, and recognition accuracy. The **delay of recognition** is defined as $t_{recog}(a_j)$ minus $t_{end}(a_{j-1})$. A negative value for the delay of recognition indicates that the recognition of an action primitive has occurred prior to the completion of a preceding action primitive. In this case, the RNN has successfully *predicted* an action primitive. Prediction of action primitives can enhance the seamlessness of the shared autonomy system.

The **delay of stable recognition** is defined as the duration between $t_{stable}(a_j)$ minus $t_{end}(a_{j-1})$. As with the delay of recognition, it is possible for the delay of stable recognition value to be negative. When considering positive values of the delay of recognition and

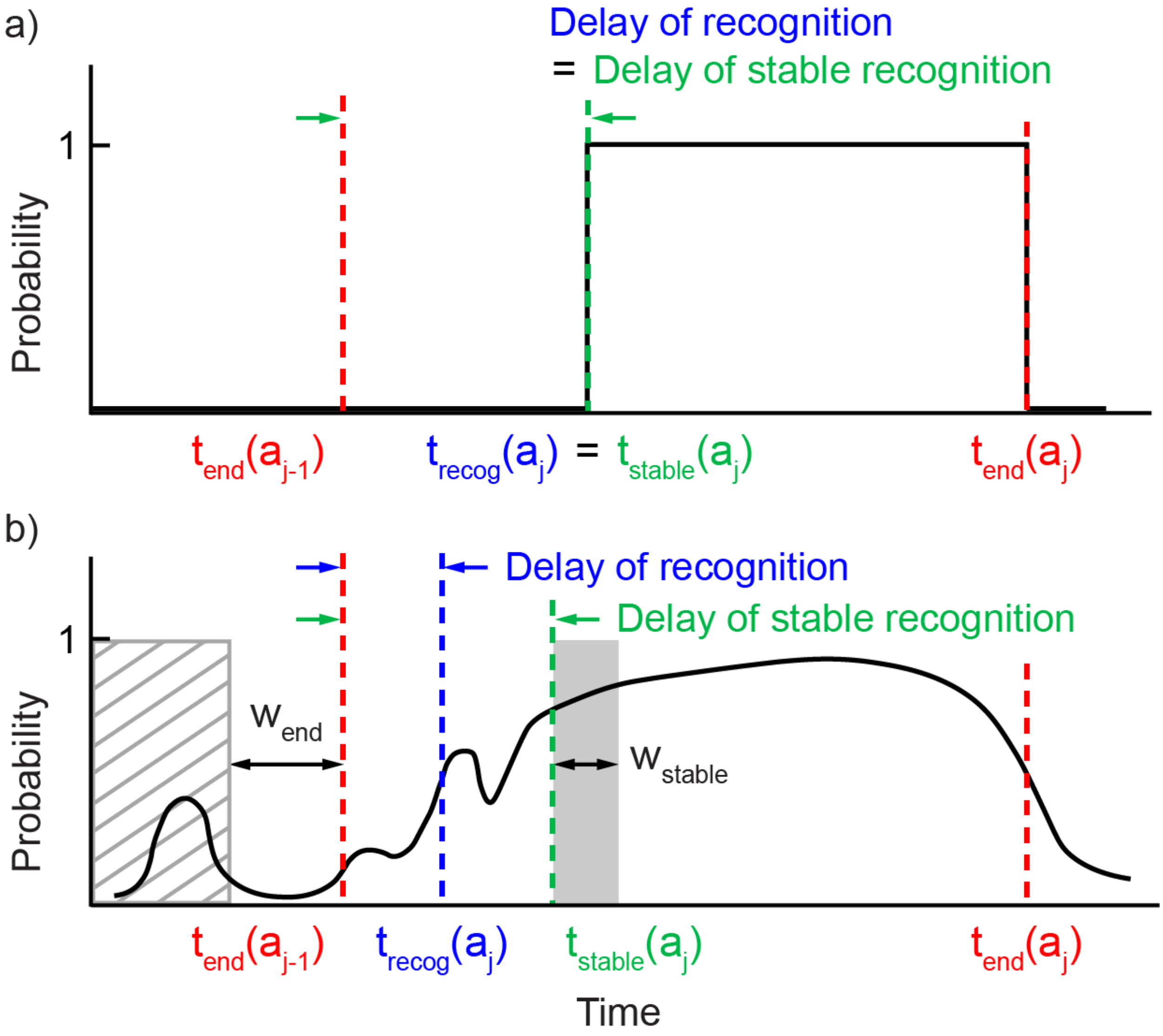


Figure 4.2: Key variables and performance metrics described in Section 4.5.2 are defined for the (a) GT mode and (b) AP mode. Since the GT mode does not utilize an intent inference model, the delay of recognition equals the delay of stable recognition. For the AP mode, the delay of stable recognition depends on the intent inference model and user-defined time windows such as W_{end} and W_{stable}. Prediction of action primitives is only possible with the AP mode.

delay of stable recognition, as the delay magnitudes decrease, the seamlessness of the shared autonomy system increases.

Recognition accuracy is defined as the proportion of time steps from $t_{end}(a_{j-1})$ to $t_{end}(a_j)$ that are correctly identified as a_j. Ground truth for each action primitive was known since all participants followed instructions to perform 10 specific action primitives in a given sequence. For the GT mode, recognition is deemed correct when a_j matches a_{lock} as determined by the locking mechanism described in section 4.4.6. For the AP mode, recognition is deemed correct when a_j matches a_{recog} as determined by eq. 4.1, which relies upon the action primitive recognition RNN.

4.5.2.3 Subjective Measures

After each trial, we adopted verbatim the questionnaire reported in [15]. Using a Likert scale ranging from 1-7, we asked participants to respond to the following statements, where 1 and 7 corresponded to "strongly disagree" and "strongly agree," respectively:

1. "I felt in control."
2. "The robot did what I wanted."
3. "I was able to accomplish the task quickly."

At the end of the experimental session, we asked participants two open-ended questions:

1. "Which control mode do you prefer and why?"
2. "Do you have any general comments for either the first or the second control mode?"

4.5.3 Specification of user-defined parameters

The implementation of the gaze-based shared autonomy framework and the performance assessment involve a number of user-defined variables. The following values were determined from preliminary studies in order to balance speed with robustness of performance.

In Section 4.4.5.1, to extract an operator's intended target position for setting down a grasped object, we set the moving filter window $w_{f,gaze}$ as 0.5 sec and the distance threshold d_r as 5 cm. In Section 4.4.4.2, to filter out noise in the real-time action primitive recognition RNN outputs, we set the moving average filter window $w_{f,recog}$ as 0.5 sec. For the "locking" mechanism, the probability thresholds p_{lock} and p_{unlock} were set as 0.7 and 0.3, respectively. Considering the 2 sec and 1.5 sec windows used for gaze triggering in [90] and [92], respectively, we set our "locking" window w_{lock} as 1.5 sec to enable a fair comparison of the GT and AP modes. In Section 4.6.2, to calculate objective measures of performance, we set the windows w_{end} and w_{stable} as 1 sec and 0.5 sec, respectively.

4.6 Results

4.6.1 Gaze Behavior

We utilized the gaze object sequence described in our prior work [30] to track each participant's gaze patterns throughout each experimental trial. The gaze object sequences were normalized temporally across all 48 trials (3 trials for each of 16 subjects) and pooled.

At a population level, the gaze behaviors with respect to gaze object sequence were not notably different between the GT and AP modes. The similarity in gaze behaviors for the GT and AP modes was a surprising finding. We considered the gaze behaviors reported in [30] and used to train the RNN in this work as “natural.” We believed the gaze fixation behaviors required of traditional gaze trigger methods to be less natural.

There are a number of reasons why the GT and AP gaze behaviors might be more similar than expected. First, the human subject experiments in [30] used to train the RNN did not involve robot hardware at all. The introduction of a robot in this work could have altered gaze behaviors [120] to the point that any differences that might have existed between the GT and AP control modes would be overshadowed. Second, regardless of the control mode, participants typically fixated on target objects as if to ensure that their intent would be correctly communicated to the robot. While this gaze fixation was not required for the AP mode, participants were not told how either control mode worked. Gaze fixation could have resulted from the brief instructions that were given to all participants regarding the experimental session as a whole: “You can let the robot know your intent by looking at the target object.” If participants believed that their gaze behaviors were in direct control of the robot, they may have focused on carefully controlling their eye movements. Third, the similarity of the GT and AP gaze behaviors could also reflect, to some extent, a participant’s lack of trust in the robot’s control algorithms.

For three out of 16 participants, the 3D gaze behaviors differed between the benchmark GT and proposed AP modes. Figure 4.3 shows the 3D gaze behavior for a representative trial for the action primitive “move mug A to mug B.” Instead of fixating on the target object mug B, three participants tracked the hand object mug A via a smooth pursuit eye movement during the AP control mode (Figure 4.3b). Despite the different gaze behaviors used by these three participants, the AP mode succeeded in identifying their intended action primitives. It is possible that the robust performance of the AP control mode resulted from the exposure of the RNN classifier to a variety of gaze behaviors during training.

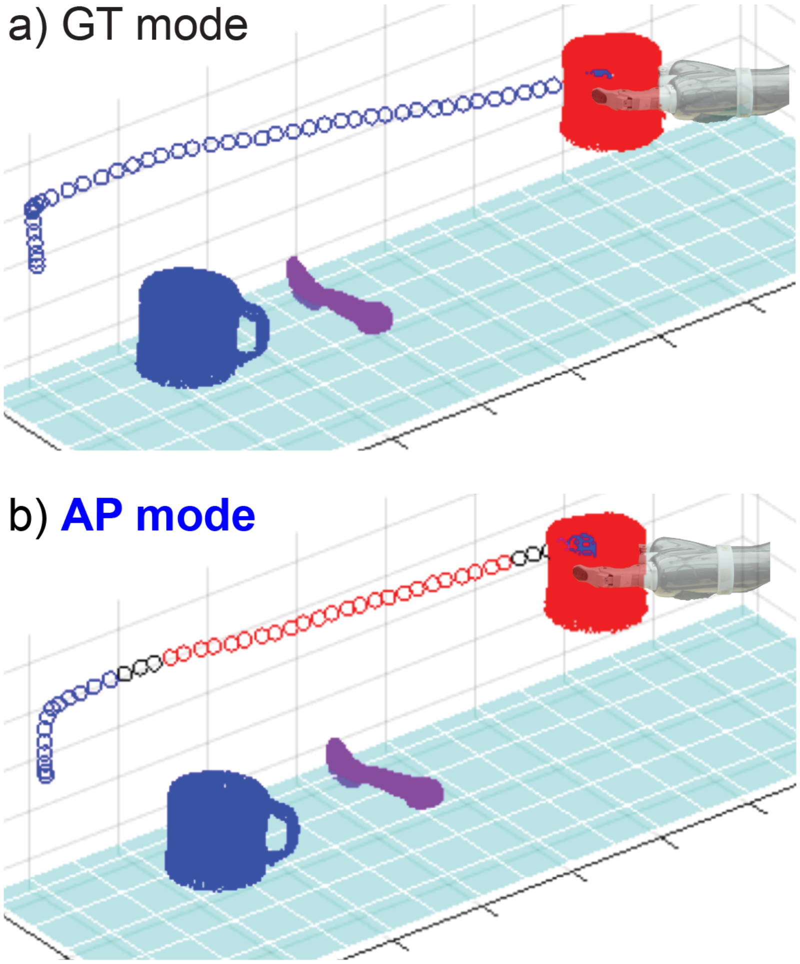


Figure 4.3: For three participants, 3D gaze behaviors differed between the GT and AP modes. The gaze object sequence for the action primitive "move mug A to mug B" is overlaid in color on the 3D end-effector path for an individual trial for the (a) GT mode and (b) AP mode. Circular markers, shown for every 100 ms, are colored according to the gaze object: mug A (red), mug B (blue), spoon (purple), and table (black).

4.6.2 Objective Measures of Performance

For each action primitive and control mode, we report the population averages for the delay of recognition, delay of stable recognition, and recognition accuracy (Table 4.1). Since all 16 participants operated the robot using both control modes, we conducted a paired t-test with a significance level of $\alpha = 0.05$.

In general, the AP mode outperformed the benchmark GT mode for all three objective measures of performance. First, we will address the delay of recognition. Out of the nine action primitives that yielded statistically significant results, five action primitives had a mean delay of recognition that was negative, indicating that prediction of action primitives had occurred. Prediction occurred for action primitives involving verbs "set down" and "move." For these two verbs, eye gaze moves toward the target object of the subsequent action primitive well in advance, which may enable the predictive capabilities of the AP mode. For the action primitive "move mug A to mug B," the prediction of the action primitive occurred as much as 0.86 sec prior to the completion of the prior action primitive.

All delay of recognition values for the GT mode were positive, indicating that prediction of action primitives was not possible with the GT mode. Furthermore, all positive delay of recognition values were smaller for the AP mode than those of the GT mode. Notably, the AP mode outperformed the GT mode by 1 sec or greater for eight out of 10 action primitives. For the two manipulate-type action primitives that were the exceptions ("stir" and "pour"), the target objects were already gazed at during the preceding action primitives involving the verb "move." As a result, the delay of recognition was less than 1 sec for both control modes.

By design, the delay of stable recognition performance metric is more strict and conservative than the delay of recognition performance metric. Thus, the delay of stable recognition values were either equal to or slightly worse than those for the delay of recognition for all action primitives and control modes. Predictive abilities were degraded by less than 0.2 sec, as in the “move spoon to mug A” case. The AP mode outperformed the GT mode by the largest margin (1.96 sec) for the second “reach mug A” action primitive in the instructed sequence (Table 4.1).

Regarding recognition accuracy, the mean recognition accuracy was statistically significantly higher for the AP mode than the GT mode for all 10 action primitives. For the AP mode, the average recognition accuracy exceeded 95% for five out of the 10 action primitives, and exceeded 85% for all 10 action primitives. The lowest recognition accuracy value was 71.4% for the GT mode as compared with 86.3% for the AP mode.

Of the three objective metrics of performance, the delay of stable recognition appeared to be the most reliable metric of seamlessness of the shared autonomy system. We highlight the delay of stable recognition results for all 10 action primitives in Figure 4.4. The mean delay of stable recognition was lower for the AP mode than the GT mode for all action primitives except for “pour,” which had a p-value of 0.11 (Table 4.1).

Predictive capabilities of the AP mode were observed for the five action primitives involving “set down” or “move” (Figure 4.4). The use of the RNN classifier for action primitive recognition resulted in earlier recognition and execution of users’ intended action primitives. By enhancing the responsiveness of the robot to gaze behaviors, without sacrificing recognition accuracy, the AP mode resulted in a more seamless gaze-based shared autonomy system than the GT mode.

The AP control mode uses an RNN model that leverages gaze object angle (GOA) and gaze object angular speed (GOAS), which are not considered by the GT mode at all. As

Table 4.1: Paired t-tests were conducted for the GT and AP modes for three objective metrics of performance. Population means are reported, with the best result for each action primitive shaded in gray and the best overall result for each performance metric indicated in bold for cases that were statistically significant ($\alpha = 0.05$).

Action Primitive	Ctrl. Mode	Objective Metrics of Performance		
		Delay of recog. (sec)	Delay of stable recog. (sec)	Recog. Acc. (%)
1. Reach mug A	GT	2.12	2.12	78.7
	AP	0.32	0.32	95.8
		$p<0.01$	$p<0.01$	$p<0.01$
2. Set down mug A	GT	1.01	1.01	71.4
	AP	-0.61	-0.60	91.3
		$p<0.01$	$p<0.01$	$p<0.01$
3. Reach spoon	GT	2.90	2.90	72.2
	AP	1.02	1.29	86.3
		$p<0.01$	$p<0.01$	$p<0.01$
4. Move spoon to mug A	GT	1.37	1.37	78.3
	AP	-0.42	-0.26	96.7
		$p<0.01$	$p=0.03$	$p=0.02$
5. Stir	GT	0.81	0.81	91.8
	AP	0.12	0.12	98.4
		$p<0.01$	$p<0.01$	$p<0.01$
6. Set down spoon	GT	1.18	1.18	72.2
	AP	-0.23	-0.20	94.2
		$p<0.01$	$p<0.01$	$p<0.01$
7. Reach mug A	GT	2.94	2.94	72.5
	AP	0.72	0.98	89.9
		$p<0.01$	$p<0.01$	$p<0.01$
8. Move mug A to mug B	GT	0.95	0.95	84.7
	AP	**-0.86**	**-0.86**	97.1
		$p<0.01$	$p<0.01$	$p<0.01$
9. Pour	GT	0.68	0.68	95.3
	AP	0.08	0.08	**98.9**
		$p=0.11$	$p=0.11$	$p=0.04$
10. Set down mug A	GT	1.47	1.47	79.5
	AP	-0.25	-0.25	94.9
		$p<0.01$	$p<0.01$	$p<0.01$

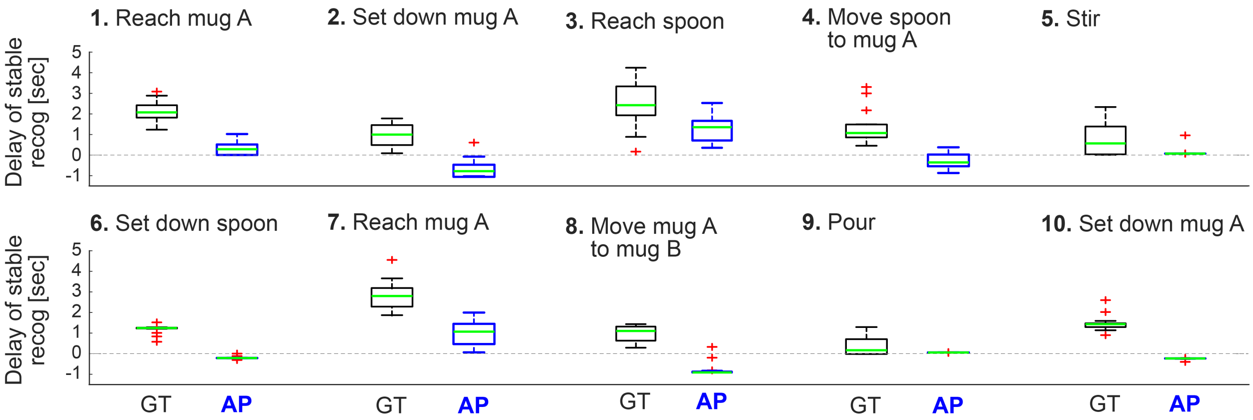


Figure 4.4: The delay of stable recognition is shown for the GT and AP modes. Each boxplot indicates the 25th, 50th (green), and 75th percentiles. The whiskers extend to the most extreme data points that are not considered outliers (red "+"), which have values that exceed 1.5 times the interquartile range from the top or bottom of the box. A negative value indicates that an action primitive has been predicted before the end of the preceding action primitive. The AP mode (blue) outperformed the GT mode (black) for all action primitives ($p \leq \alpha = 0.05$) except for "pour."

reported in our prior study [71], the use of GOA and GOAS as input features to the RNN can decrease the observational latency for recognizing action primitives. The features GOA and GOAS may encode the tendency of the gaze vector to approach an object once the eyes start to move, thereby providing intent-relevant information even before the gaze vector intersects with the target object.

4.6.3 Subjective Measures of Performance

4.6.3.1 Post-trial Survey

Figure 4.5 summarizes average participant responses to the post-trial surveys described in Section 4.5.2 for each of the control modes. A paired t-test ($\alpha = 0.05$) was conducted in order to compare participants' views for the benchmark GT and proposed AP control modes. For all three survey statements, there was a statistically significant difference between the Likert scale responses ($p < 0.01$). In each case, the AP control mode outperformed the GT mode. For the statement "I felt in control," the mean (standard deviation) Likert scale response was 6.2 (0.5) for the AP mode and 5.8 (0.8) for the GT mode. For the statement "The robot did what I wanted," the mean (standard deviation) Likert scale response was 6.5 (0.5) for the AP mode and 5.9 (0.8) for the GT mode. The largest difference in mean values was observed for the statement "I was able to accomplish the task quickly." In this case, the mean (standard deviation) Likert scale response was 6.1 (0.6) for the AP mode and 5.1 (1.0) for the GT mode.

4.6.3.2 Post-experiment Interview

As described in Section 4.5.1, half of the participants experienced the GT mode first and half of the participants experienced the AP mode. For clarity, we refer to the GT and AP modes using brackets instead of as "first" and "second" control modes, which each participant referenced in their interview responses.

In the post-experiment interview, 14 out of 16 participants expressed a preference for the AP mode (see the Supplemental Video for 1st and 3rd person perspectives of a representative trial). Representative comments about the AP mode are listed below:

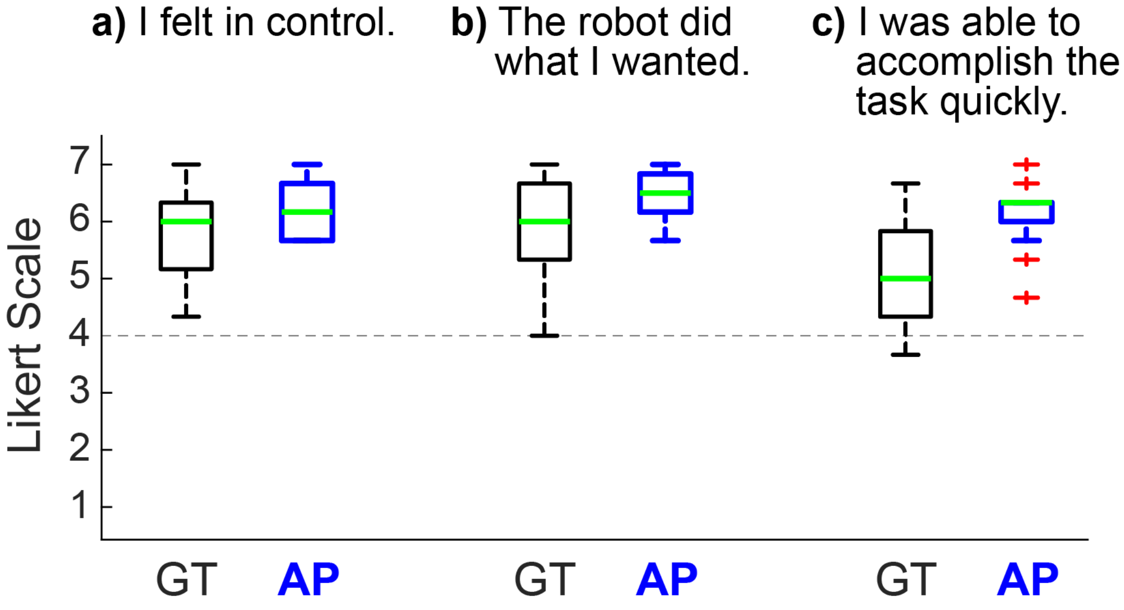


Figure 4.5: Likert scale survey results are shown, where 1 and 7 indicate “strongly disagree” and “strongly agree,” respectively. Each boxplot indicates the 25th, 50th (green), and 75th percentiles. The whiskers extend to the most extreme data points that are not considered outliers (red “+”), which have values that exceed 1.5 times the interquartile range from the top or bottom of the box. The AP mode (blue) outperformed the GT mode (black) for all three statements ($p < 0.01$).

- *"I don't have to look at one object or position for a long time like a few seconds, and the movement [of the AP mode] is smoother."*

- *"It just seems there's smooth tracking. The [AP] process is pretty fast and seamless. It is pretty obvious that the [GT] control mode is slower to respond than the [AP]."*

- *"When I was using [AP], it was more responsive, and the action is pretty smooth. There's no pause in the middle."*

- *"The [AP] control mode was more fluent. In the [GT] control mode, it didn't feel like when I looked at it, the robot is following, so I felt less control."*

Two participants preferred the GT mode over the AP mode:

- *"Although the [GT] control mode was slower, there was not any confusion. You looked at the cup, and after a couple of seconds, it picked up the cup. It took some time, but it would do it, so you don't have to worry about correcting. The other one [AP] feels a bit twitchy like it's very responsive."*

- *"For the [AP] control mode, I felt like the control was more on the robot side instead of the human side. I enjoyed it initially because I felt like I can rely on the robot to accomplish each of these tasks very accurately in a laboratory environment. Still, if I use the system in real life, there will be unpredictable variables, and I will appreciate it if the robot can pause and wait for my confirmation through eyes like what the [GT] control mode did."*

According to the Likert scale survey and the post-experiment survey, most participants reported that the AP mode was more seamless than the GT mode. For the GT mode, participants perceived pauses between the robot's execution of their intended action primitives.

For the AP mode, participants reported that the robot was more responsive to their eye movements. No subjects commented on any speed-accuracy trade-off with the AP mode.

4.6.4 Limitations and Future Work

Despite a general satisfaction with the AP mode, two participants expressed skepticism about the ability of the gaze-based shared autonomy system to accurately recognize their intent in a more visually cluttered environment. Such concerns might be assuaged by enhancing the transparency of the system and conveying to participants what the robot has inferred and plans to do next. Alonso and Puente highlighted the critical nature of transparency for shared autonomy systems in a review paper [132]. They described how transparency could improve system performance, reduce human errors, and build human trust in human-robot systems.

Zeng et al. introduced augmented reality (AR) feedback into a gaze-based shared autonomy system [125] in which participants controlled a robot by fixating on objects on a screen. The screen featured a 3rd person video on which color-coded annotations were overlaid to highlight the gazed object, aperture of the gripper, impending actions, etc. The AR feedback improved the efficiency of the system and reduced participants' cognitive load during the robotic grasping and lifting processes. However, participants looked at a 2D screen rather than the actual, 3D physical world, as in this work. Our results might be improved further through the use of an AR headset to increase the system's transparency. For instance, we could project verb and target object information and/or the planned robotic trajectory.

Our gaze-based shared autonomy framework could also be improved by developing a library with a greater variety of action primitive candidates, as would be needed for the diverse set of activities of daily living required by individuals with upper limb impairment.

Currently, the action primitive recognition thread labels each time step as one of the four verbs (reach, move, set down, manipulate). Although these four verb classes are generic enough to serve as building blocks for complex actions, important verbs such as “feed” [133] are not included. Modifications to the current action primitive recognition framework would be required, as a participant’s mouth could not be identified as a “target object” during self-feeding. Instead, action primitives related to feeding and drinking might require the triggering of pre-planned trajectories once the utensil or cup were ready to be brought to one’s mouth. Alternatively, information from a 3rd person camera perspective could be used to supplement that of the 1st person view of the user [133, 134] to improve the accuracy and safety of feeding and drinking trajectories.

4.7 Conclusion

We developed a novel gaze-based shared autonomy framework to assist with activities of daily living. Utilizing a pre-trained recurrent neural network [71], the system can recognize, and often predict, an operator’s intended action primitives using 3D gaze-related features. The system can localize objects in real-time, dynamically plan collision-free trajectories to reach, move, manipulate, and set down everyday objects. Through both objective and subjective metrics of performance, we demonstrated that our AP control mode, which leverages a gaze-based action primitive recognition model, can outperform the conventional gaze-triggered control mode. Borne out by statistical analyses and participant surveys and interviews, the AP mode enabled a more seamless gaze-based shared autonomy system than the GT mode. The system can serve as a foundation for further enhancements to system transparency through augmented reality and system adaptability through the expansion of the verb library used for action primitive recognition.

CHAPTER 5

Summary and Conclusion

Eye tracking offers a non-verbal, non-invasive, intuitive method for human operators to communicate their intent to robots. However, it can be challenging to identify useful gaze-related features, recognize intent through gaze behaviors, and integrate gaze-based intent recognition algorithms into collaborative human-robot systems. The work presented in this dissertation has provided new methodologies to create 3D gaze saliency maps for gaze behavior analysis, recognize human intent at the subtask and action primitive levels, and incorporate action recognition algorithms into shared autonomy systems with real robots.

5.1 Contributions

3D gaze saliency maps: Using reconstructed 3D gaze vectors, we created high spatial resolution 3D gaze saliency maps by assigning RGB colors to point clouds obtained from 3D scans of objects in the benchmark Yale-CMU-Berkeley (YCB) Object Set [135]. The gaze saliency maps appeared to encode action-relevant information at the subtask and action unit level. Unlike 2D gaze saliency maps that are constructed from a specific camera perspective, 3D gaze saliency maps enable gaze behavior analyses from a variety of 3D perspectives.

Novel subtask recognition method using gaze object sequences: We used the gaze object sequence (GOS) to capture information about the identity of objects in concert with the temporal sequence in which the objects were visually regarded. We used dynamic time warping barycentric averaging to create a population-based set of gaze object sequences that were characteristic of subtasks. We demonstrated recognition of subtasks by comparing its GOS with a characteristic GOS using a dynamic time warping Euclidean distance metric. We showed that the GOS can be used to achieve high recognition accuracy values and is a promising feature for action recognition.

Novel action primitive recognition method using 3D gaze-related features: We defined an action primitive as a triplet comprised of a verb, target object, and hand object. We trained a long short-term memory recurrent neural network to recognize a verb and target object, and then tested the trained network on three different activities. Using a non-specific approach to indexing objects in the workspace, we observed a modest level of generalizability of the action primitive classifier across activities, including those for which the classifier was not trained. We demonstrated that the novel input features of gaze object angle and its rate of change were especially useful for accurately recognizing action primitives and reducing the observational latency of the classifier. The classifier and the novel gaze-related features can be used to recognize intent in shared autonomy control schemes for human-robot systems.

Implementation of a gaze-based shared autonomy system on a real robot for activities of daily living: We built a shared autonomy framework capable of 3D gaze reconstruction, real-time intent recognition, object localization, obstacle avoidance, and dynamic trajectory planning. The assistive system can be used for activities of daily living that require capabilities such as "reach-to-grasp," "move," "set-down," "pour," and "stir." We developed an "action prediction" (AP) control mode by leveraging our prior pre-trained RNN-based intent prediction algorithm. Through experiments with real robots and a set of objective and subjective performance evaluation metrics, we demonstrated that the

AP control mode results in more seamless human-robot collaboration than a state-of-the-art "gaze trigger" control mode, and that participants often preferred the AP control mode.

5.2 Future Work

5.2.1 Enhance robot transparency by enabling bidirectional communication in human-robot systems

Transparency in human-robot systems is the observability and predictability of the robotic behavior [132]. In a transparent system, operators can easily understand what the system is doing, why, and what it will do next. Studies have demonstrated that transparency helps to improve system performance and build human-robot trust [136].

Although our gaze-based shared autonomy system presented in Chapter 4 is capable of real-time intent recognition, and even prediction, the current system is not sufficiently transparent. While the robot can estimate an operator's intent through their gaze behaviors, operators are not aware of the robot's classifier results – neither the action recognized nor the timing of a stable recognition. To address this, we could enable bidirectional communication in human-robot interactions by utilizing augmented reality (AR). For example, we could substitute the eye tracker with AR glasses capable of eye tracking (e.g. Microsoft Hololens headset). We could overlay a semi-transparent image of the robot's target pose onto the real workspace in order to communicate the robot's planned movement to operators as soon as a stable action recognition has been achieved through gaze tracking [137, 138, 139].

As reported in Chapter 4 and Figure 4.3, participants' gaze behaviors were not natural for either the action prediction mode or the gaze trigger mode. Participants visually fixated on target objects in order to ensure that their intent would be correctly conveyed to the robot. We believe that the integration of an AR-based bidirectional communication channel between the human and robot could enable operators to place more trust in robot autonomy, and possibly result in more natural gaze behaviors during human-robot collaboration.

5.2.2 Measure cognitive burden and trust using physiological sensors

While we have demonstrated that our proposed action prediction control mode can make a human-robot shared autonomy system more seamless, we did not evaluate the operator's cognitive burden or trust in the assistive robot. To further advance our proposed shared autonomy control scheme, it is necessary to quantify how intuitive it is to control the robot and to what extent participants trust the robot, and preferably to do so in real-time. Both cognitive burden and trust will affect gaze behaviors and, in turn, the overall performance and quality of the human-robot collaboration.

Previous studies measured cognitive burden using physiological signals such as galvanic skin conductance [140], electrocardiogram (ECG) signals [141], and pupil dilation [7], as well as facial expressions [142]. While not reported in this dissertation, the experiments described in Chapter 4 were conducted with participants outfitted with galvanic skin conductance and electrocardiogram sensors. The analysis of those data are intended as future work.

Human trust plays an important role in human-robot systems, especially in our proposed gaze-based shared autonomy system. A lack of trust could significantly affect participants' eye gaze behaviors. For instance, when they do not trust a robot, operators might repeatedly check the robot's state for visual cues associated with problematic robot behaviors (e.g. impending collisions) [143]. Such untrusting gaze behaviors could result in action recognition errors by the classifier if these gaze behavior characteristics were not used during model training and/or as trust evolves over time. Numerous factors could concurrently affect human-robot trust including the reliability of the robot, participants' understanding of the system, participants' prior experience with robots, the consequences of failure, etc [144]. As future work, one can attempt to quantify an operator's trust in a robot using physiological measurements, such as galvanic skin conductance and electrocardiogram signals. By incorporating additional real-time data on the physiological state of the operator, a robotic system could adjust its recognition and planning algorithms according to the operator's level of trust and further improve the quality of the human-robot collaboration.

5.2.3 Increase the robot's versatility

A long-term goal of this work is to advance gaze-based shared autonomy control schemes so that individuals with upper limb impairment can more naturally control robots that assist with activities of daily living (ADLs). In Chapter 4, we demonstrated the ability of a human-robot shared autonomy system to reach, move, set down, pour, and stir with objects from the kitchen. Due to the complexity and variety of ADLs, however, the shared autonomy system of the future will need to be even more versatile.

With that philosophy in mind, the action primitive method proposed in Chapter 3 was designed to assign generalizable labels using one of four common verb classes (reach, move, set down, and manipulate). The verb class "manipulate" includes many subclasses of manipulate-type verbs, such as "pour" and "stir." A shared autonomy system could provide more specific assistance by recognizing additional subclasses of verbs. One idea is to create a look-up table based on object affordances [57]. For example, the action primitive triplet of (verb=manipulate, TO=mug, HO=pitcher) is associated with the verb subclass "pour." However, the triplet (verb=manipulate, TO=pitcher, HO=spoon) is associated with both verb subclasses "stir" and "scoop." An alternative approach could be to leverage 3D gaze saliency maps, as described in Chapter 2, to facilitate the classification of verb subclasses since action primitives are activity-driven and the distribution of gaze fixations can be considerably affected by object affordance [40, 30].

While the four verb classes are intended to be easily generalized to different objects and task contexts, some ADL-related verbs, such as feeding, are missing. One could use the verb class “move” for a feeding task since a utensil is moved from a container to one’s mouth. Using the framework described in Chapter 3, this would require the training of a human mouth as a “target object.” Alternatively, one could also expand the verb class set to include ADL-specific verbs. Either way, additional data collection, classifier training, and testing on real robots are needed.

Finally, our current shared autonomy framework is designed for object-oriented tasks that have easily defined start and end poses and trajectories. However, tasks requiring the continuous control of the end-effector or handheld tool have not been addressed. Examples of open-ended ADL-related tasks include cleaning a countertop using a sponge or cleaning oneself with a napkin. Operators could benefit from a shared autonomy system that can transition seamlessly between clearly defined object manipulation tasks and less constrained continuous end-point control of a manipulator. Some studies have already investigated gaze-driven, continuous robot control on planar surfaces, such as controlling a manipulator to write [89] and draw [145], and controlling the movement of a wheelchair [13]. We envision an organic integration of the continuous end-effector control mode and the discrete object manipulation mode in which the shared autonomy system can identify operators’ intended mode through natural eye movements and provide corresponding assistance under the recognized mode.

REFERENCES

[1] M. K. Eckstein, B. Guerra-Carrillo, A. T. Miller Singley, and S. A. Bunge, "Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development?" *Developmental Cognitive Neuroscience*, vol. 25, pp. 69–91, Jun. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1878929316300846

[2] A. J. Calder, A. D. Lawrence, J. Keane, S. K. Scott, A. M. Owen, I. Christoffels, and A. W. Young, "Reading the mind from eye gaze," *Neuropsychologia*, vol. 40, no. 8, pp. 1129–1138, Jan. 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0028393202000088

[3] H. Zamani, A. Abas, and M. K. M.Amin, "Eye Tracking Application on Emotion Analysis for Marketing Strategy," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 8, no. 11, pp. 87–91, Dec. 2016, number: 11. [Online]. Available: https://jtec.utem.edu.my/jtec/article/view/1415

[4] P. T. Huddleston, B. K. Behe, C. Driesener, and S. Minahan, "Inside-outside: Using eye-tracking to investigate search-choice processes in the retail environment," *Journal of Retailing and Consumer Services*, vol. 43, pp. 85–93, Jul. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969698917306227

[5] L. Eger, "How people acquire knowledge from a web page: An eye tracking study," *Knowledge Management & E-Learning: An International Journal*, vol. 10, no. 3, pp. 350–366, Nov. 2018, number: 3. [Online]. Available: http://kmel-journal.org/ojs/index.php/online-publication/article/view/14

[6] R. S. A. Khan, G. Tien, M. S. Atkins, B. Zheng, O. N. M. Panton, and A. T. Meneghetti, "Analysis of eye gaze: Do novice surgeons look at the same location as expert surgeons during a laparoscopic operation?" *Surgical Endoscopy*, vol. 26, no. 12, pp. 3536–3540, Dec. 2012. [Online]. Available: https://doi.org/10.1007/s00464-012-2400-7

[7] K. Rayner, K. H. Chace, T. J. Slattery, and J. Ashby, "Eye Movements as Reflections of Comprehension Processes in Reading," *Scientific Studies of Reading*, vol. 10, no. 3, pp. 241–255, Jul. 2006, publisher: Routledge _eprint: https://doi.org/10.1207/s1532799xssr1003_3. [Online]. Available: https://doi.org/10.1207/s1532799xssr1003_3

[8] E. Granholm, R. F. Asarnow, A. J. Sarkin, and K. L. Dykes, "Pupillary responses index cognitive resource limitations," *Psychophysiology*, vol. 33, no. 4, pp. 457–461, 1996, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8986.1996.tb01071.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.1996.tb01071.x

[9] M. M. Bradley, L. Miccoli, M. A. Escrig, and P. J. Lang, “The pupil as a measure of emotional arousal and autonomic activation,” *Psychophysiology*, vol. 45, no. 4, pp. 602–607, 2008, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8986.2008.00654.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.2008.00654.x

[10] J. P. Hansen, A. Alapetite, I. S. MacKenzie, and E. Møllenbach, “The use of gaze to control drones,” in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA ’14. New York, NY, USA: Association for Computing Machinery, Mar. 2014, pp. 27–34. [Online]. Available: https://doi.org/10.1145/2578153.2578156

[11] M. Yu, Y. Lin, D. Schmidt, X. Wang, and Y. Wang, “Human-Robot Interaction Based on Gaze Gestures for the Drone Teleoperation,” *Journal of Eye Movement Research*, vol. 7, pp. 1–14, Sep. 2014.

[12] L. Yuan, C. Reardon, G. Warnell, and G. Loianno, “Human Gaze-Driven Spatial Tasking of an Autonomous MAV,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1343–1350, Apr. 2019, conference Name: IEEE Robotics and Automation Letters.

[13] L.-A. Raymond, M. Piccini, M. Subramanian, O. Pavel, and A. Faisal, “Natural Gaze Data Driven Wheelchair,” Tech. Rep., Jan. 2018, company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article. [Online]. Available: https://www.biorxiv.org/content/10.1101/252684v1

[14] S. Li, X. Zhang, F. J. Kim, R. Donalisio da Silva, D. Gustafson, and W. R. Molina, “Attention-Aware Robotic Laparoscope Based on Fuzzy Interpretation of Eye-Gaze Patterns,” *Journal of Medical Devices*, vol. 9, no. 4, p. 041007, Aug. 2015.

[15] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, “Shared autonomy via hindsight optimization for teleoperation and teaming,” *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, Jun. 2018, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364918776060

[16] C. Yu and D. H. Ballard, “Understanding human behaviors based on eye-head-hand coordination,” in *International Workshop on Biologically Motivated Computer Vision*, H. Bülthoff, C. Wallraven, S. Lee, and T. Poggio, Eds. Springer, 2002, pp. 611–619.

[17] W. Yi and D. Ballard, “Recognizing behavior in hand-eye coordination patterns,” *International Journal of Humanoid Robotics*, vol. 06, no. 03, pp. 337–359, Sep. 2009. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0219843609001863

[18] A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding Egocentric Activities," in *Proceedings of the IEEE International Conference on Computer Vision*. Barcelona, Spain: IEEE, 2011, pp. 407–414.

[19] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in *European Conference on Computer Vision*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer, 2012, pp. 314–327.

[20] A. Fathi and J. M. Rehg, "Modeling actions through state changes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA: IEEE, 2013, pp. 2579–2586.

[21] A. Behera, D. C. Hogg, and A. G. Cohn, "Egocentric activity monitoring and recovery," in *Asian Conference on Computer Vision*, K. Lee, Y. Matsushita, J. Rehg, and Z. Hu, Eds. Berlin, Heidelberg: Springer, 2012, pp. 519–532.

[22] K. Matsuo, K. Yamada, S. Ueno, and S. Naito, "An Attention-based Activity Recognition for Egocentric Video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, USA, 2014, pp. 551–556.

[23] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA: IEEE, 2016, pp. 1894–1903.

[24] B. Soran, A. Farhadi, and L. Shapiro, "Action Recognition in the Presence of One Egocentric and Multiple Static Cameras," in *Computer Vision – ACCV 2014*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds. Cham: Springer International Publishing, 2015, vol. 9007, pp. 178–193, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-319-16814-2_12

[25] H. Admoni and S. Srinivasa, "Predicting User Intent Through Eye Gaze for Shared Autonomy," in *Proc AAAI Fall Symposium Series: Shared Autonomy in Research and Practice*, Arlington, VA, Nov. 2016, pp. 298–303.

[26] C.-M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration." IEEE, Mar. 2016, pp. 83–90. [Online]. Available: http://ieeexplore.ieee.org/document/7451737/

[27] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-Loop Optimization of Shared Autonomy in Assistive Robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 247–254, Jan. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7518989/

[28] S. Li and X. Zhang, “Implicit Intention Communication in Human–Robot Interaction Through Visual Behavior Studies,” *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 437–448, Aug. 2017, conference Name: IEEE Transactions on Human-Machine Systems.

[29] S. Fuchs and A. Belardinelli, “Gaze-Based Intention Estimation for Shared Autonomy in Pick-and-Place Tasks,” *Frontiers in Neurorobotics*, vol. 15, p. 647930, Apr. 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8085393/

[30] A. Haji Fathaliyan, X. Wang, and V. J. Santos, “Exploiting Three-Dimensional Gaze Tracking for Action Recognition During Bimanual Manipulation to Enhance Human–Robot Collaboration,” *Frontiers in Robotics and AI*, vol. 5, pp. 1–15, 2018.

[31] B. D. Argall, “Turning assistive machines into assistive robots,” M. Razeghi, E. Tournié, and G. J. Brown, Eds., San Francisco, California, United States, Jan. 2015, p. 93701Y. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2085352

[32] R. S. Johansson, G. Westling, A. Bäckström, and J. R. Flanagan, “Eye-Hand Coordination in Object Manipulation,” *Journal of Neuroscience*, vol. 21, no. 17, pp. 6917–6932, Sep. 2001, publisher: Society for Neuroscience Section: ARTICLE. [Online]. Available: https://www.jneurosci.org/content/21/17/6917

[33] S. J. Lederman and R. L. Klatzky, “Hand Movements: A Window Into Haptic Object Recognition,” *Cognitive Psychology*, vol. 19, no. 3, pp. 342–368, Jul. 1987. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/3608405

[34] M. Land, N. Mennie, and J. Rusted, “The roles of vision and eye movements in the control of activities of daily living,” *Perception*, vol. 28, no. 11, pp. 1311–1328, 1999.

[35] D. Kanoulas and M. Vona, “Bio-inspired rough terrain contact patch perception,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on.* Hong Kong, China: IEEE, 2014, pp. 1719–1724.

[36] S. Li, X. Zhang, and J. Webb, “3D-Gaze-based Robotic Grasping through Mimicking Human Visuomotor Function for People with Motion Impairments,” *IEEE Transactions on Biomedical Engineering*, pp. 1–1, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7870669/

[37] A. Behera, M. Chapman, A. G. Cohn, and D. C. Hogg, “Egocentric activity recognition using histograms of oriented pairwise relations,” in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 2. Lisbon, Portugal: IEEE, 2014, pp. 22–30.

[38] T.-H.-C. Nguyen, J.-C. Nebel, and F. Florez-Revuelta, "Recognition of Activities of Daily Living with Egocentric Vision: A Review," *Sensors*, vol. 16, no. 1, p. 72, Jan. 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/1/72

[39] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[40] A. Belardinelli, O. Herbort, and M. V. Butz, "Goal-oriented gaze strategies afforded by object interaction," *Vision Research*, vol. 106, pp. 47–57, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698914002818

[41] A. Haji Fathaliyan, X. Wang, S. Bazargan, and V. Santos, "Hand-object kinematics and gaze fixation during bimanual tasks," in *Proc Ann Mtg American Society of Biomechanics*, Boulder, CO, Aug. 2017.

[42] R. C. Oldfield, "The assessment and analysis of handedness: the Edinburgh inventory," *Neuropsychologia*, vol. 9, no. 1, pp. 97–113, 1971. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0028393271900674

[43] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols," *arXiv preprint arXiv:1502.03143*, 2015. [Online]. Available: http://arxiv.org/abs/1502.03143

[44] E. H. Spriggs, F. De La Torre, and M. Hebert, "Temporal segmentation and activity classification from first-person sensing," in *Proc of the IEEE Conf on Computer Vision and Pattern Recognition Workshops*. Miami Beach, Florida: IEEE, 2009, pp. 17–24.

[45] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, Eds., *Principles of neural science*, 4th ed. New York: McGraw-Hill, Health Professions Division, 2000.

[46] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," *Behavior Research Methods*, vol. 42, no. 1, pp. 188–204, Feb. 2010. [Online]. Available: https://link.springer.com/article/10.3758/BRM.42.1.188

[47] J.-Y. Bouguet, "Camera Calibration Toolbox for MATLAB," Oct. 2015. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[48] T. MathWorks, "Single Camera Calibration App," 2017. [Online]. Available: https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html

[49] R. E. Morrison and K. Rayner, "Saccade size in reading depends upon character spaces and not visual angle," *Perception & Psychophysics*, vol. 30, no. 4, pp. 395–396, Jul. 1981. [Online]. Available: https://link.springer.com/article/10.3758/BF03206156

[50] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos, “Gait recognition using dynamic time warping,” in *IEEE Workshop on Multimedia Signal Processing*, Siena, Italy, 2004, pp. 263–266.

[51] D. M. Gavrila and L. S. Davis, “Towards 3-D model-based tracking and recognition of human movement: a multi-view approach,” in *International Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, 1995, pp. 272–277.

[52] F. Petitjean, “MATLAB function for ”DBA: Averaging time series consistently with Dynamic Time Warping”,” Nov. 2016. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/47483-fpetitjean-dba

[53] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. New York: Cambridge University Press, 2008, oCLC: ocn190786122.

[54] A. Harvey, J. N. Vickers, R. Snelgrove, M. F. Scott, and S. Morrison, “Expert surgeon’s quiet eye and slowing down: expertise differences in performance and quiet eye duration during identification and dissection of the recurrent laryngeal nerve,” *The American Journal of Surgery*, vol. 207, no. 2, pp. 187–193, Feb. 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0002961013005679

[55] C.-a. Moulton, G. Regehr, L. Lingard, C. Merritt, and H. MacRae, “Slowing Down to Stay Out of Trouble in the Operating Room: Remaining Attentive in Automaticity:,” *Academic Medicine*, vol. 85, no. 10, pp. 1571–1577, Oct. 2010. [Online]. Available: http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage&an=00001888-201010000-00013

[56] M. Hayhoe and D. Ballard, “Eye movements in natural behavior,” *Trends in Cognitive Sciences*, vol. 9, no. 4, pp. 188–194, Apr. 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661305000598

[57] J. J. Gibson, “The theory of affordances,” in *Perceiving, Acting, and Knowing: Towards an Ecological Psychology*, R. Shaw and J. Bransford, Eds. Hoboken, NJ: John Wiley & Sons Inc., 1977, pp. 127–143.

[58] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Kruger, O. Kroemer, J. Peters, and J. Piater, “Learning object-specific grasp affordance densities,” in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*. Shanghai, China: IEEE, 2009, pp. 1–7.

[59] A. Leclercq, S. Akkouche, and E. Galin, “Mixing Triangle Meshes and Implicit Surfaces in Character Animation,” in *Computer Animation and Simulation 2001: Proceedings of the Eurographics Workshop in Manchester, UK, September 2–3, 2001*, N. Magnenat-Thalmann and D. Thalmann, Eds. Vienna: Springer Vienna, 2001, pp. 37–47. [Online]. Available: https://doi.org/10.1007/978-3-7091-6240-8_4

[60] J. Pearson and S. M. Kosslyn, Eds., *Mental Imagery*, ser. Frontiers Research Topics. Frontiers Media SA, 2013. [Online]. Available: http://www.frontiersin.org/books/Mental_Imagery/188

[61] M. F. Land and M. Hayhoe, "In what ways do eye movements contribute to everyday activities?" *Vision Research*, vol. 41, no. 25, pp. 3559–3565, Nov. 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S004269890100102X

[62] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on.* Madrid, Spain: IEEE, 2014, pp. 527–534.

[63] R. Luo, R. Hayne, and D. Berenson, "Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces," *Autonomous Robots*, Jul. 2017. [Online]. Available: http://link.springer.com/10.1007/s10514-017-9655-8

[64] C. Morato, K. N. Kaipa, B. Zhao, and S. K. Gupta, "Toward Safe Human Robot Collaboration by Using Multiple Kinects Based Real-time Human Tracking," *Journal of Computing and Information Science in Engineering*, vol. 14, no. 1, p. 011006, Jan. 2014. [Online]. Available: http://computingengineering.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4025810

[65] S. S. Srinivasa, D. Berenson, M. Cakmak, A. Collet, M. R. Dogar, A. D. Dragan, R. A. Knepper, T. Niemueller, K. Strabala, and M. V. Weghe, "Herb 2.0: Lessons learned from developing a mobile manipulator for the home," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2410–2428, 2012.

[66] Q. Jenkins and X. Jiang, "Measuring trust and application of eye tracking in human robotic interaction," in *IIE Annual Conference. Proceedings.* Cancun, Mexico: Institute of Industrial and Systems Engineers (IISE), 2010, p. 1.

[67] G. Westerfield, A. Mitrovic, and M. Billinghurst, "Intelligent Augmented Reality Training for Motherboard Assembly," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 157–172, Mar. 2015. [Online]. Available: http://link.springer.com/10.1007/s40593-014-0032-x

[68] F. E. Truitt, C. Clifton, A. Pollatsek, and K. Rayner, "The Perceptual Span and the Eye-Hand Span in Sight Reading Music," *Visual Cognition*, vol. 4, no. 2, pp. 143–161, Jun. 1997. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/713756756

[69] X. Anguera, R. Macrae, and N. Oliver, "Partial sequence matching using an unbounded dynamic time warping algorithm," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.* Dallas, TX, USA: IEEE, 2010, pp. 3582–3585.

[70] L. Montesano and M. Lopes, “Learning grasping affordances from local visual descriptors,” in *2009 IEEE 8th International Conference on Development and Learning*, Jun. 2009, pp. 1–6.

[71] X. Wang, A. Haji Fathaliyan, and V. J. Santos, “Toward Shared Autonomy Control Schemes for Human-Robot Systems: Action Primitive Recognition Using Eye Gaze Features,” *Frontiers in Neurorobotics*, vol. 14, p. 567571, Oct. 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2020.567571/full

[72] S. S. Groothuis, S. Stramigioli, and R. Carloni, “Lending a helping hand: toward novel assistive robotic arms,” *IEEE Robotics Automation Magazine*, vol. 20, no. 1, pp. 20–29, Mar. 2013, conference Name: IEEE Robotics Automation Magazine.

[73] B. Driessen, H. Evers, and J. v Woerden, “MANUS—a wheelchair-mounted rehabilitation robot,” *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 215, no. 3, pp. 285–290, Jan. 2001. [Online]. Available: http://journals.pepublishing.com/content/pr833615804485th/

[74] V. Maheu, J. Frappier, P. S. Archambault, and F. Routhier, “Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities,” in *2011 IEEE International Conference on Rehabilitation Robotics*. Zurich: IEEE, Jun. 2011, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/5975397/

[75] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola, and R. Sukthankar, “Exploring the Trade-off Between Accuracy and Observational Latency in Action Recognition,” *International Journal of Computer Vision*, vol. 101, no. 3, pp. 420–436, Feb. 2013. [Online]. Available: https://doi.org/10.1007/s11263-012-0550-7

[76] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm,” *Nature*, vol. 485, no. 7398, pp. 372–375, May 2012, number: 7398 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/nature11076

[77] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann, “Using gesture and speech control for commanding a robot assistant,” in *11th IEEE International Workshop on Robot and Human Interactive Communication Proceedings*, Berlin, Germany, Sep. 2002, pp. 454–459.

[78] L. Bi, A. >. Feleke, and C. Guan, “A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration,” *Biomedical Signal Processing and Control*, vol. 51, pp. 113–127, May 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1746809419300473

[79] L. Bi, X. Fan, and Y. Liu, “EEG-Based Brain-Controlled Mobile Robots: A Survey,” *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 2, pp. 161–176, Mar. 2013.

[80] A. F. Salazar-Gomez, J. DelPreto, S. Gil, F. H. Guenther, and D. Rus, “Correcting robot mistakes in real time using EEG signals,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, May 2017, pp. 6570–6577.

[81] Z. C. Chao, Y. Nagasaka, and N. Fujii, “Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey,” *Frontiers in Neuroengineering*, vol. 3, 2010. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fneng.2010.00003/full

[82] W. Wang, J. L. Collinger, A. D. Degenhart, E. C. Tyler-Kabara, A. B. Schwartz, D. W. Moran, D. J. Weber, B. Wodlinger, R. K. Vinjamuri, R. C. Ashmore, J. W. Kelly, and M. L. Boninger, “An Electrocorticographic Brain Interface in an Individual with Tetraplegia,” *PLOS ONE*, vol. 8, no. 2, p. e55344, Feb. 2013. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0055344

[83] S. E. Ghobadi, O. E. Loepprich, F. Ahmadov, K. Hartmann, O. Loffeld, and J. Bernshausen, “Real Time Hand Based Robot Control Using Multimodal Images,” *IAENG International Journal of Computer Science*, vol. 35, no. 4, pp. 110–121, 2008. [Online]. Available: http://www.iaeng.org/IJCS/issues_v35/issue_4/IJCS_35_4_08.pdf

[84] J. L. Raheja, R. Shyam, U. Kumar, and P. B. Prasad, “Real-Time Robotic Hand Control Using Hand Gestures,” in *2010 Second International Conference on Machine Learning and Computing*, Bangalore, India, Feb. 2010, pp. 12–16.

[85] M. M. Hayhoe, A. Shrivastava, R. Mruczek, and J. B. Pelz, “Visual memory and motor planning in a natural task,” *Journal of Vision*, vol. 3, no. 1, pp. 6–6, Jan. 2003. [Online]. Available: https://jov.arvojournals.org/article.aspx?articleid=2158157

[86] C.-S. Lin, C.-W. Ho, W.-C. Chen, C.-C. Chiu, and M.-S. Yeh, “Powered wheelchair controlled by eye-tracking system.” *Optica Applicata*, vol. 36, pp. 401–412, 2006. [Online]. Available: http://opticaapplicata.pwr.edu.pl/article.php?id=2006230401

[87] P. S. Gajwani and S. A. Chhabria, “Eye motion tracking for wheelchair control,” *International Journal of Information Technology*, vol. 2, no. 2, pp. 185–187, 2010. [Online]. Available: http://csjournals.com/IJITKM/PDF%203-1/2.pdf

[88] 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*, vol. 64, no. 12, pp. 2824–2835, 2017.

[89] S. Dziemian, W. W. Abbott, and A. A. Faisal, “Gaze-based teleprosthetic enables intuitive continuous control of complex robot arm use: Writing & drawing,” in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2016, pp. 1277–1282.

[90] M.-Y. Wang, A. A. Kogkas, A. Darzi, and G. P. Mylonas, “Free-View, 3D Gaze-Guided, Assistive Robotic System for Activities of Daily Living,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain: IEEE, 2018, pp. 2355–2361.

[91] H. Zeng, Y. Shen, X. Hu, A. Song, B. Xu, H. Li, Y. Wang, and P. Wen, “Semi-Autonomous Robotic Arm Reaching With Hybrid Gaze–Brain Machine Interface,” *Frontiers in Neurorobotics*, vol. 13, 2020, publisher: Frontiers. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnbot.2019.00111/full?utm_source=S-TWT&utm_medium=SNET&utm_campaign=ECO_FNINS_XXXXXXXX_auto-dlvrit

[92] A. Shafti, P. Orlov, and A. A. Faisal, “Gaze-based, Context-aware Robotic System for Assisted Reaching and Grasping,” in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, May 2019, pp. 863–869, iSSN: 2577-087X, 1050-4729.

[93] T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, Nov. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314206001263

[94] F. Lv and R. Nevatia, “Recognition and Segmentation of 3-D Human Action Using HMM and Multi-class AdaBoost,” in *Computer Vision – ECCV*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer, 2006, pp. 359–372.

[95] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, Jun. 2012, pp. 1290–1297.

[96] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, Jun. 2014, pp. 588–595. [Online]. Available: http://ieeexplore.ieee.org/document/6909476/

[97] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, Jun. 2015, pp. 1110–1118. [Online]. Available: http://ieeexplore.ieee.org/document/7298714/

[98] Y. Li, Z. Ye, and J. M. Rehg, "Delving into egocentric actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 287–295.

[99] Y. Zhang, "Edinburgh Handedness Inventory (revised)," 2012. [Online]. Available: http://zhanglab.wikidot.com/handedness

[100] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* San Juan, Puerto Rico, USA: IEEE, Jun. 1997, pp. 1106–1112, iSSN: 1063-6919.

[101] M. A. A. Haseeb and R. Parasuraman, "Wisture: RNN-based Learning of Wireless Signals for Gesture Recognition in Unmodified Smartphones," *arXiv:1707.08569 [cs]*, Jul. 2017, arXiv: 1707.08569. [Online]. Available: http://arxiv.org/abs/1707.08569

[102] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference for Learning Representations*, San Diego, CA, USA, 2015, pp. 1–13. [Online]. Available: https://dblp.org/db/conf/iclr/iclr2015.html

[103] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," in *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI)*, Las Vegas, Nevada, USA, 2000, pp. 111–117.

[104] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, "Learning from imbalanced data sets with weighted cross-entropy function," *Neural Processing Letters*, pp. 1–13, 2019.

[105] D. C. Wells, "The Mode Filter: A Nonlinear Image Processing Operator," in *Instrumentation in Astronomy III*, vol. 0172. Tucson, AZ, USA: International Society for Optics and Photonics, May 1979, pp. 418–421. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/0172/0000/The-Mode-Filter-A-Nonlinear-Image-Processing-Operator/10.1117/12.957111.short

[106] M. F. Land, "Eye movements and the control of actions in everyday life," *Progress in Retinal and Eye Research*, vol. 25, no. 3, pp. 296–324, May 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350946206000036

[107] G. Hoffman, "Evaluating Fluency in Human–Robot Collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, Jun. 2019, conference Name: IEEE Transactions on Human-Machine Systems.

[108] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *2011 International Conference on Computer Vision*, Barcelona, Spain, Nov. 2011, pp. 1036–1043.

[109] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 619–635.

[110] A. Furnari and G. Farinella, "What Would You Expect? Anticipating Egocentric Actions With Rolling-Unrolling LSTMs and Modality Attention," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 6251–6260. [Online]. Available: https://ieeexplore.ieee.org/document/9008264/

[111] S. Sudhakaran, S. Escalera, and O. Lanz, "LSTA: Long Short-Term Attention for Egocentric Action Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 9946–9955. [Online]. Available: https://ieeexplore.ieee.org/document/8954401/

[112] M. Liu, S. Tang, Y. Li, and J. Rehg, "Forecasting Human-Object Interaction: Joint Prediction of Motor Attention and Actions in First Person Video," *arXiv:1911.10967 [cs]*, Jul. 2020, arXiv: 1911.10967. [Online]. Available: http://arxiv.org/abs/1911.10967

[113] S. Schaal, "Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics," in *Adaptive Motion of Animals and Machines*, H. Kimura, K. Tsuchiya, A. Ishiguro, and H. Witte, Eds. Tokyo, Japan: Springer, 2006, pp. 261–280. [Online]. Available: https://doi.org/10.1007/4-431-31381-8_23

[114] B. Velichkovsky, A. Sprenger, and P. Unema, "Towards gaze-mediated interaction: Collecting solutions of the "Midas touch problem"," in *Human-Computer Interaction INTERACT '97: IFIP TC13 International Conference on Human-Computer Interaction, 14th–18th July 1997, Sydney, Australia*, ser. IFIP — The International Federation for Information Processing, S. Howard, J. Hammond, and G. Lindgaard, Eds. Boston, MA: Springer US, 1997, pp. 509–516. [Online]. Available: https://doi.org/10.1007/978-0-387-35175-9_77

[115] C.-M. Huang and A. L. Thomaz, "Joint Attention in Human-Robot Interaction," in *2010 AAAI Fall Symposium Series*, Nov. 2010. [Online]. Available: https://www.aaai.org/ocs/index.php/FSS/FSS10/paper/view/2173

[116] R. Eidenberger and J. Scharinger, "Active perception and scene modeling by planning with probabilistic 6D object poses," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 1036–1043, iSSN: 2153-0866.

[117] J. H. Kim, K. Abdel-Malek, Z. Mi, and K. Nebel, "Layout Design using an Optimization-Based Human Energy Consumption Formulation," SAE International, Warrendale, PA, SAE Technical Paper 2004-01-2175, Jun. 2004, iSSN: 0148-7191,

2688-3627. [Online]. Available: https://www.sae.org/publications/technical-papers/content/2004-01-2175/

[118] D. Ognibene and G. Baldassare, "Ecological Active Vision: Four Bioinspired Principles to Integrate Bottom–Up and Adaptive Top–Down Attention Tested With a Simple Camera-Arm Robot," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 1, pp. 3–25, Mar. 2015, conference Name: IEEE Transactions on Autonomous Mental Development.

[119] D. Ognibene and Y. Demiris, "Towards Active Event Recognition," in *Twenty-Third International Joint Conference on Artificial Intelligence*, Jun. 2013. [Online]. Available: https://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6705

[120] R. M. Aronson, T. Santini, T. C. Kübler, E. Kasneci, S. Srinivasa, and H. Admoni, "Eye-Hand Behavior in Human-Robot Shared Manipulation," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: Association for Computing Machinery, Feb. 2018, pp. 4–13. [Online]. Available: https://doi.org/10.1145/3171221.3171287

[121] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, Jun. 2013. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364913490324

[122] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How Autonomy Impacts Performance and Satisfaction: Results From a Study With Spinal Cord Injured Subjects Using an Assistive Robot," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 1, pp. 2–14, Jan. 2012, conference Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans.

[123] A. Bilyea, N. Seth, S. Nesathurai, and H. A. Abdullah, "Robotic assistants in personal care: A scoping review," *Medical Engineering & Physics*, vol. 49, pp. 1–6, Nov. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350453317301777

[124] J. Ramos and S. Kim, "Humanoid Dynamic Synchronization Through Whole-Body Bilateral Feedback Teleoperation," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 953–965, Aug. 2018, conference Name: IEEE Transactions on Robotics.

[125] H. Zeng, Y. Wang, C. Wu, A. Song, J. Liu, P. Ji, B. Xu, L. Zhu, H. Li, and P. Wen, "Closed-Loop Hybrid Gaze Brain-Machine Interface Based Robotic Arm Control with Augmented Reality Feedback," *Frontiers in Neurorobotics*, vol. 11, 2017, publisher: Frontiers. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnbot.2017.00060/full

[126] Y. Wang, G. Xu, A. Song, B. Xu, H. Li, C. Hu, and H. Zeng, "Continuous Shared Control for Robotic Arm Reaching Driven by a Hybrid Gaze-Brain Machine Interface," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4462–4467, iSSN: 2153-0866.

[127] Y.-S. L.-K. Cio, M. Raison, C. L. Ménard, and S. Achiche, "Proof of Concept of an Assistive Robotic Arm Control Using Artificial Stereovision and Eye-Tracking," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 12, pp. 2344–2352, Dec. 2019, conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[128] C.-M. Huang, S. Andrist, A. Sauppé, and B. Mutlu, "Using gaze patterns to predict task intent in collaboration," *Frontiers in Psychology*, vol. 6, Jul. 2015. [Online]. Available: http://journal.frontiersin.org/Article/10.3389/fpsyg.2015.01049/abstract

[129] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "MoveIt! Task Constructor for Task-Level Motion Planning," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 190–196, iSSN: 2577-087X.

[130] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *1985 IEEE International Conference on Robotics and Automation Proceedings*, vol. 2, Mar. 1985, pp. 500–505.

[131] Q. Zhu, Y. Yan, and Z. Xing, "Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing," in *Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2. Jian, China: IEEE, Oct. 2006, pp. 622–627. [Online]. Available: http://ieeexplore.ieee.org/document/4021735/

[132] V. Alonso and P. de la Puente, "System Transparency in Shared Autonomy: A Mini Review," *Frontiers in Neurorobotics*, vol. 12, 2018, publisher: Frontiers.

[133] D. Park, Y. Hoshi, H. P. Mahajan, H. K. Kim, Z. Erickson, W. A. Rogers, and C. C. Kemp, "Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned," *Robotics and Autonomous Systems*, vol. 124, p. 103344, Feb. 2020.

[134] D. Park, Y. Hoshi, and C. C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018, conference Name: IEEE Robotics and Automation Letters.

[135] B. C. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research," 2015. [Online]. Available: http://ycb-benchmarks.s3-website-us-east-1.amazonaws.com/

[136] E. Rosen, D. Whitney, M. Fishman, D. Ullman, and S. Tellex, “Mixed Reality as a Bidirectional Communication Interface for Human-Robot Interaction,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 11 431–11 438, iSSN: 2153-0866.

[137] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, “Communicating Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays,” *arXiv:1708.03655 [cs]*, Aug. 2017, arXiv: 1708.03655. [Online]. Available: http://arxiv.org/abs/1708.03655

[138] E. Ruffaldi, F. Brizzi, F. Tecchia, and S. Bacinelli, “Third Point of View Augmented Reality for Robot Intentions Visualization,” in *Augmented Reality, Virtual Reality, and Computer Graphics*, ser. Lecture Notes in Computer Science, L. T. De Paolis and A. Mongelli, Eds. Cham: Springer International Publishing, 2016, pp. 471–478.

[139] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, “Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 5628–5635, iSSN: 2577-087X.

[140] N. Nourbakhsh, Y. Wang, F. Chen, and R. A. Calvo, “Using galvanic skin response for cognitive load measurement in arithmetic and reading tasks,” in *Proceedings of the 24th Australian Computer-Human Interaction Conference*, ser. OzCHI ’12. New York, NY, USA: Association for Computing Machinery, Nov. 2012, pp. 420–423. [Online]. Available: https://doi.org/10.1145/2414536.2414602

[141] R. Xiong, F. Kong, X. Yang, G. Liu, and W. Wen, “Pattern Recognition of Cognitive Load Using EEG and ECG Signals,” *Sensors*, vol. 20, no. 18, p. 5122, Jan. 2020, number: 18 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5122

[142] A. Pecchinenda and M. Petrucci, “Emotion Unchained: Facial Expression Modulates Gaze Cueing under Cognitive Load,” *PLOS ONE*, vol. 11, no. 12, p. e0168111, Dec. 2016, publisher: Public Library of Science. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0168111

[143] R. M. Aronson and H. Admoni, “Gaze for error detection during human-robot shared manipulation,” June 2018.

[144] Z. R. Khavas, S. R. Ahmadzadeh, and P. Robinette, “Modeling Trust in Human-Robot Interaction: A Survey,” in *Social Robotics*, ser. Lecture Notes in Computer Science, A. R. Wagner, D. Feil-Seifer, K. S. Haring, S. Rossi, T. Williams, H. He, and S. Sam Ge, Eds. Cham: Springer International Publishing, 2020, pp. 529–541.

[145] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, “A Novel Robotic System for Painting with Eyes,” in *Advances in Italian Mechanism Science*, ser. Mechanisms and Machine Science, V. Niola and A. Gasparetto, Eds. Cham: Springer International Publishing, 2021, pp. 191–199.