AN ECOLOGICALLY VALID REFERENCE FRAME FOR PERSPECTIVE INVARIANT ACTION RECOGNITION

A Thesis

by

Berkay Bayram

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
August 2021

AN ECOLOGICALLY VALID REFERENCE FRAME FOR PERSPECTIVE INVARIANT ACTION RECOGNITION

Approved by:

Professor Erhan Öztop, Advisor
Department of Computer Science
Özyeğin University

Associate Professor Emre Uğur
Department of Computer Engineering
Boğaziçi University

Assistant Professor Furkan Kıraç
Department of Computer Science
Özyeğin University

Date Approved: 05 August 2021

To my grandmother, parents, sister and friends

ABSTRACT

In robotics, objects and body parts can be represented in various coordinate frames to ease computation. In biological systems, body or body part centered coordinate frames have been proposed as possible reference frames that the brain uses for interacting with the environment. Coordinate transformations are standard tools in robotics and can facilitate perspective invariant action recognition and action prediction based on observed actions of other agents. Although it is known that human adults can do explicit coordinate transformations, it is not clear whether this capability is used for recognizing and understanding the actions of others. Mirror neurons, found in the ventral premotor cortex of macaque monkeys, seem to undertake action understanding in a perspective invariant way, which may rely on lower level perceptual mechanisms. To this end, in this paper, we propose a novel reference frame that is ecologically plausible and can sustain basic action understanding and mirror function. We demonstrate the potential of this representation by simulation of an upper body humanoid robot with an action repertoire consisting of push, poke, move-away, bring-to-mouth, bring-left and bring-right actions. The simulation experiments indicate that the representation is suitable for action recognition and effect prediction in a perspective invariant way, and thus can be deployed as an artificial mirror system for robotic applications.

ÖZETÇE

Robotikte, nesneler ve vücut parçaları, hesaplamayı kolaylaştırmak için çeşitli koordinat sistemlerinde temsil edilebilir. Biyolojik sistemlerde, vücut veya vücut parçası merkezli koordinat sistemleri, beynin çevre ile etkileşim için kullandığı olası referans sistemleri olarak önerilmiştir. Koordinat dönüşümleri robotikte standart araçlardır ve diğer ajanların gözlemlenen eylemlerine dayalı olarak perspektiften bağımsız eylem tanıma ve eylem tahminini kolaylaştırabilir. Yetişkin insanların kolaylıkla koordinat dönüşümleri yapabildikleri bilinmesine rağmen, bu yeteneğin başkalarının eylemlerini tanımak ve anlamak için kullanılıp kullanılmadığı açık değildir. Makak maymunlarının ventral premotor korteksinde bulunan ayna nöronları, daha düşük seviyeli algısal mekanizmalara dayanabilen, perspektiften bağımsız bir şekilde eylem anlayışını üstleniyor gibi görünmektedir. Bu amaçla, bu çalışmada, ekolojik olarak var olabilecek olan, temel eylem anlayışını ve ayna nöronu işlevini sürdürebilen yeni bir referans sistemi öneriyoruz. Bu temsilin potansiyelini, itme, dürtme, uzaklaştırma, ağzına getirme ve sağa ve sola taşıma eylemlerinden oluşan bir eylem repertuarına sahip bir üst vücut insansı robotun simülasyonu ile gösteriyoruz. Simülasyon deneyleri, temsilin, perspektiften bağımsız bir şekilde eylem tanıma, objeler üzerindeki etki tahmini için uygun olduğunu ve bu nedenle robotik uygulamalar için yapay bir ayna nöronu sistemi olarak konuşlandırılabileceğini göstermektedir.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Prof. Dr. Erhan Öztop for his guidance, support, and patience.

I would also like to thank my thesis committee members Dr. Emre Uğur and Dr. Furkan Kıraç for their invaluable feedback and valuable time.

Finally, I am grateful to my family, my mother Hale Bayram, father Mehmet Bayram, sister Ece Bayram and grandmother Gülten Akar for their unconditional support and love. It would not be possible to conclude my study without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

How infants learn to develop the skill to detect equivalence (parity) between their own action and the observed ones is still an open scientific question [1], which is interesting not only for biological sciences but also for robotics. One idea proposed for the development of this skill is that infants first learn how to perform 'coordinate transformation', which is the skill of rotation and translating of a 3D object so as to predict how it would look from the infant's own perspective. In robotics this is a common operation; one can obtain the self-perspective 3D pose of an object given in any arbitrary coordinate frame by using a straightforward transformation. Although adult humans seem to have this ability as an explicit skill (usually called metal rotation) [2], it is unknown how this develops and whether it is directly related to action understanding. To be concrete, it is unknown whether it is the precursor of perspective invariant action understanding although often it is accepted as such.

Animals necessarily are aware of the effects of gravity and thus it is reasonable to postulate that one of the axes they would use to assess action and effect is formed by the direction of gravity. In fact, it is shown that the monkey brain uses gravity direction dependent representations of object orientations [3]. For primates who are equipped with dexterous hand useability, it is critical to monitor moving hands for error correction in the case of self-execution and for predicting others' action goals for appropriate social behavior. This intuition is supported by neurophysiological findings showing that a special brain area in the superior temporal cortex is evolved for hand movement detection regardless of the actor [4]. Therefore it makes sense that the prediction of the effect of an action is defined with respect to the movement of the hand.

1.1 Motivation

Combining these two pieces of information, in this thesis we propose a novel reference frame that is ecologically plausible, and present our results on its viability to sustain action understanding and mirror neuron function. We call this the Action Reference Frame or Action Frame (AF) in short. We further propose that predictive learning can use AF to represent the data generated by self-action and self-observation so that generalization to others is possible, without rejecting the possibility of parallel predictive systems that employ other reference frames. To complete the reference frame definition, it is also necessary to state the origin of the AF. Two main possibilities exist: either AF is placed on the moving hand or on the object that is the target of the action. Although, in the multiple object scenarios, the latter might bring ambiguity during the initial portions of action, in the current report we took this (latter) approach due to the more straightforward analysis it allows.

In addition to the biological relevance of this idea, we are interested in implementing an action recognition capability for a self-learning robotic system. To this end, all of the work is implemented on an upper-body humanoid robot, TOROBO[1] simulation, and the processing steps are kept at a feasible level for physical robot deployment. Also to show that the results obtained are not due to the high fidelity information available from the simulator, a depth camera-based color-coded object and hand detection system is integrated into the prediction network. Overall, the results indicate that the proposed AF based prediction system can undertake action recognition and effect prediction in a perspective invariant manner. Thus, it may serve as a mirror neuron system architecture that is amenable to robotic implementation.

[1] Tokyo Robotics, `https://robotics.tokyo/products/torobo/`

1.2 Thesis Outline

The thesis is organized as follows. Chapter 2 gives information about the evidence found on Mirror Neurons and their possible functions in the monkey and human brains together with the learning system employed in this work (Artificial Neural Networks (ANNs)). Chapter 3 briefly describes the previous work in the literature on action recognition in computer vision and robotics, additionally, computational models of mirror neurons are given. In Chapter 4, the simulation environment, action designs, the construction of AF, and learning details are given. Moreover, in Chapter 5, the results of the first experiments with using only a single object are presented, later in Chapter 6, we also present the results for multiple objects under two different settings as the fixed and variable object positions for both EF and AF. Finally, Chapter 7 concludes this thesis with a brief discussion on limitations and future directions.

CHAPTER II

BACKGROUND

2.1 Mirror Neurons

2.1.1 Introduction

Neurons in the ventral premotor area F5 of macaque monkeys have been found to be activated during goal directed actions (i.e. grasping, holding, tearing) [5, 6, 7]. Also, as a result of the following research similar neurons have been encountered in the inferior parietal area (PFG of monkeys) [8, 9]. These are called Mirror Neurons and they discharge both when an action is executed and the action is observed [6].

In area F5 there are two kinds of visuomotor neurons, these are described as canonical neurons [10] and mirror neurons. While canonical neurons are sensitive to the shape and size of three-dimensional objects that one sees [11], mirror neurons encode the motor acts of goal directed actions [6, 7]. Important information about these neurons is that they operate as a matching mechanism for grasp types (i.e. precision, power grasp) on presented objects and observation of mouth and hand motor acts executing an action. Thus, creating the same activation (the same state in a sense) on the observer's brain and allowing them to automatically understand other's intentions. Additionally to visual inputs, these neurons are also observed to be active when the sound of an action related to the action is heard by the monkey(i.e. cracking sound of a peanut) [12] supporting the hypothesis that these neurons encode the goal of actions performed by others.

Similar to macaque monkeys mirror neuron regions also have been encountered in the human brain using brain imaging techniques such as PET, EGG, MEG, fMRI, etc. [13, 14, 15].

Moreover, based on this facility of the monkey brain region becoming active both while executing and observing actions, it has been hypothesized that Mirror Neurons play an important role in action understanding [16, 17], imitation [18, 19] and the evolution of language [20, 21].

2.1.2 The Mirror Neuron Circuit in Macaque Brain

The mirror neuron circuitry consists of different sub-regions of the macaque monkey. Neurons located in the superior temporal sulcus (STS) in the macaque monkey brain are responsive to the observation of actions executed by others[2]. STS is connected to the ventral premotor cortex (area F5) and the inferior parietal lobule (IPL), specifically, anterior intraparietal (AIP) and inferior parietal area (PFG) in IPL (see Figure 1).

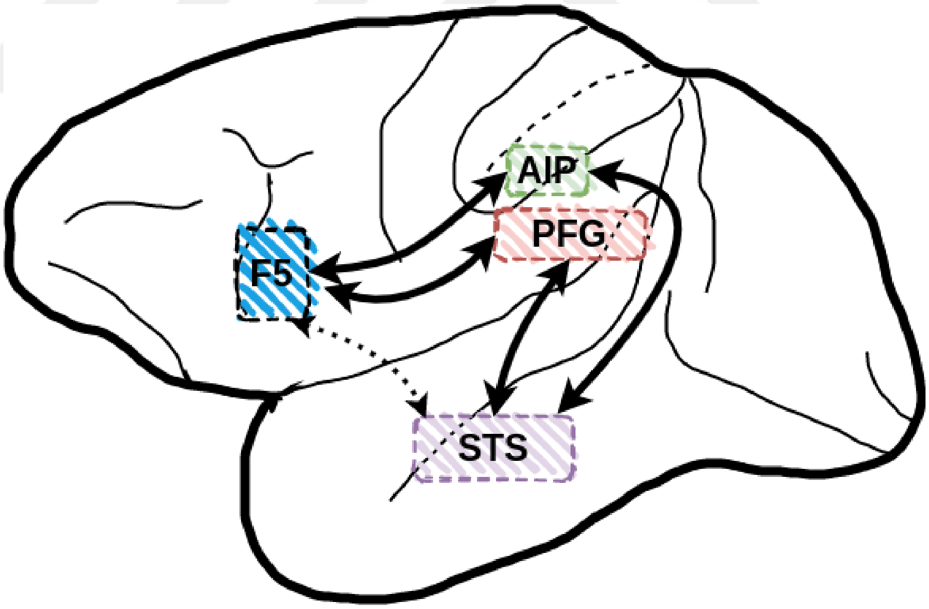

Figure 1: Simplified illustration of macaque brain and connections of the mirror neuron circuitry adapted from [27].

Experiments [22] of tracing the connections and activations of macaque monkey brain using fMRI imaging technique revealed that connects of lower and upper bank of STS is strong with PFG and AIP respectively. Neurons in STS do not discharge to active movements rather they are sensitive to the actions with definitive means (i.e. grasping). STS was observed to play an important role in understanding actions by responding to objects or actions in a view-invariant way [23].

2.1.3 The Mirror Neurons and View Invariance

Generally, mirror neurons are perspective invariant action understanding however new studies showed that some neurons get activated depending on the perspective. In one study [24] concerning view-invariant facility of mirror neurons, scientists have observed are F5 activations of macaque brain while the monkey was subject to observe action (grasping) carried out by self and another monkey that is self (0 degrees), side view (90 degrees) and a frontal view (180 degrees), what they found is that most of the neurons were active during the action which was carried out by self and some of the neurons were active during the observation of different perspectives (90 and 180 degrees). This may be connected to first learning from self-executed actions and later gaining the ability to understand others' actions with the activation of different regions on the mirror neurons system.

Also, it is hypothesized that low order view-dependent neurons together create the view independence based on the discovery that STS was found to be responsive to varying perspectives of the head[25]. Both view-dependent and independent neurons in the ventral premotor cortex (area F5) encode the goals of motor acts. Based on the previous statement when the bidirectional structure of STS[26] (also see Figure 1) is taken into account, it is argued that the view-dependent mirror neurons may play a role in understanding the perspective for observed actions[27]. Furthermore, it is thought that view-dependent mirror neurons may be beneficial to action understanding, by reinforcing the visual signal encoded by STS which is dependent on the perspective.

2.2 Coordinate Frames/Representations in the Brain

Besides the existence of mirror neuron mechanism for perspective invariant action understanding, an additional line of research lends strong support to the proposed Action Frame concept. Experimental evidence shows that the brain uses multiple spatial representations and reference frames for action production [28] and location recognition [29]. These reference frames are used to encode spatial information with respect to a collection of reference frames, including the egocentric ones such as head, eyes, hand, and body, as well as the allocentric ones such as the position of an object in attention. Finally, more specific support to the proposed Action Frame is provided by the recent findings showing that the gravity direction is well incorporated in the parietal representation of object orientations [3] suggesting that reference frames incorporating the gravity direction exist in the brain.

2.3 Learning

2.3.1 Issues in Artificial Neural Networks

The idea of creating an Artificial Neural Network(ANN) similar to the human brain is theorized in the early 1940s by McCulloch and Pitts[30]. Later, Hebb[31] studied the relationship between the environment and behavior. He suggested that the repeated stimulus increased the connectivity of neural pathways. With the increased computational power Hopfield[32] came up with an associative memory structure similar to the human brain. With his research, he showed that the usefulness of such systems. The evolution of ANNs continued with the use of the backpropagation algorithm[33] to this day and they show that they are useful agents which are applicable to real-world problems.

Artificial Neural Networks are basically universal function approximators. A simple conventional ANN consists of layers and can have an arbitrary number of layers. A layer consists of neurons that are connected to every other neuron in the next layer.

The first layer of the network is called the input layer and the last layer is called the output layer. Layers between input and output layers are called hidden layers (see Figure 2). Increasing the depth of a network increases its learning capabilities, however, introduces issues to the training process. Networks with many layers are called deep neural networks.

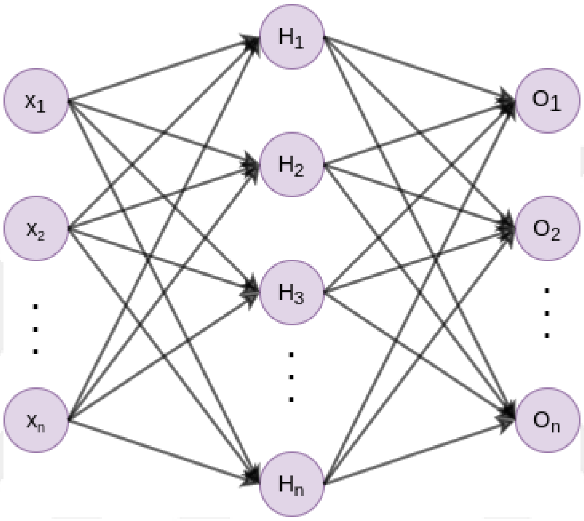

Figure 2: A simple artificial neural network with three layers.

2.3.1.1 Neuron

A neuron in ANNs inspired by nature (i.e. the human brain) is the smallest member of an ANN. This artificial neuron is a mathematical function in its essence. Input to an artificial neuron receives one or more inputs and these could be any arbitrary number, using smaller numbers (such as [0, 1] or [-1, 1]) found to be work better with the increased depth of the neural network. The inputs are weighted with their connection, fed to the neuron, and summed. While training these connections (weights) are calibrated ever so slightly to minimize the difference between neuron's output and target value. Furthermore, artificial neurons also have a bias value that is trainable. After all weights are summed and bias added neuron's output goes in to the activation function sigma ($\sigma(\cdot)$) (see equation 1). This calculation forms the output of a neuron and is delegated to the next neuron if connected to any.

$$y = \sigma(\sum_{i=1}^{n}(w_i * x_i) + b) \tag{1}$$

2.3.1.2 Activation Functions

Activation functions used in ANNs are non-linear functions that receive a single neuron output. The purpose of using non-linearities at the neuron output is to be able to stack many neurons one after another (sequential connections). Without non-linear functions, neural networks are not able to learn complex representations since connections between neurons become simple multiplication operations and eliminate hidden layers.

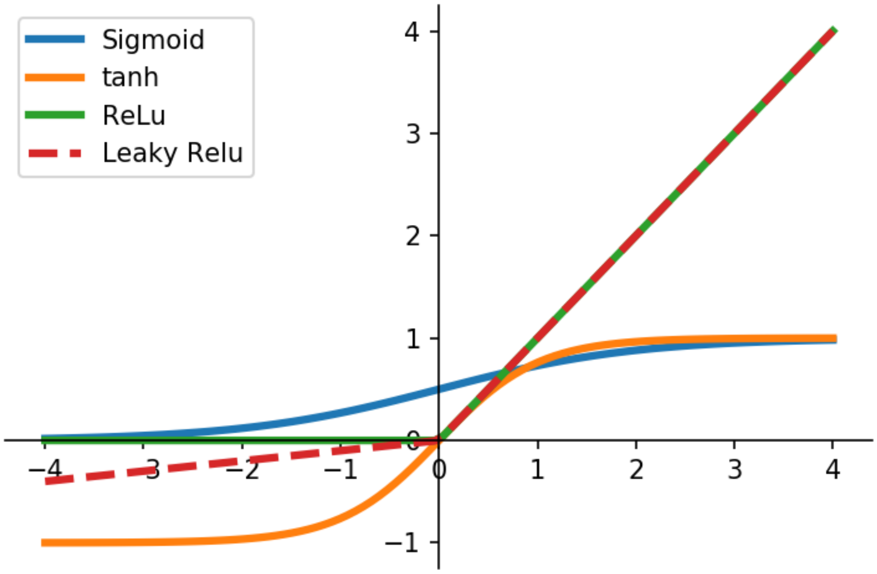

Figure 3: Showing different activation functions.

There are different non-linear functions which are used as activation functions in neural networks such as the Sigmoid[34], tanh[35], Rectified Linear Units (ReLU)[36], Leaky ReLu[37] etc. (see Figure 3 for the functions and Figure 4 for their derivatives). Among these activation functions, ReLU is widely used in current literature since it is both computationally faster and allows better gradient flow thru the network relative to the others. One limitation of ReLU is that when neurons' output is smaller than or equal to zero no learning takes place since gradients would also be zero and error cannot propagate thru the network. Leaky ReLU encounters this issue by allowing the gradient to flow when the input is smaller or equal to zero with a small slope line. However, this comes with a computation overhead. In this study, we used ReLU in our ANN as the choice of the activation function.

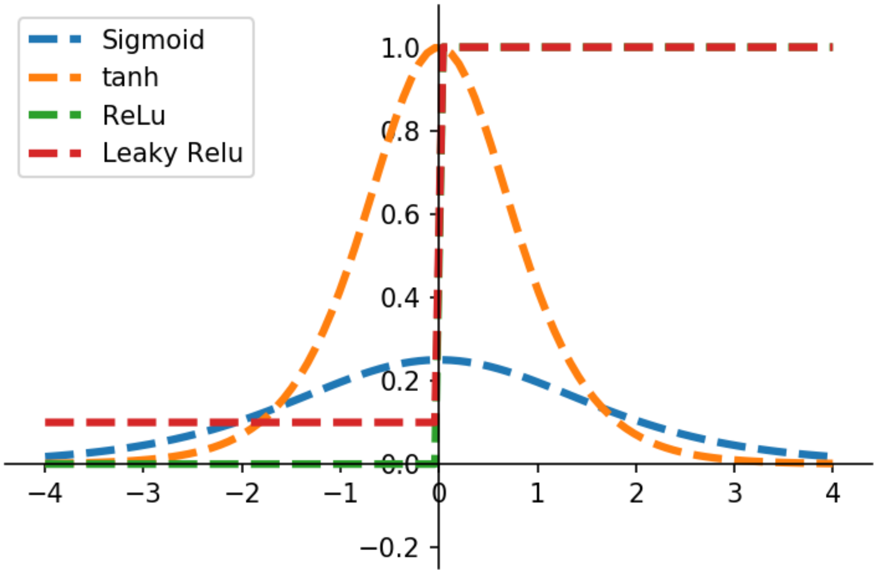

Figure 4: Showing the derivative of different activation functions.

2.3.1.3 Gradient Descent

Gradient Descent is an iterative differentiation algorithm to minimize the error (the distance between the network output and the target) of an artificial neural network. In each step of the algorithm small steps (determined by the learning rate) are taken towards the opposite direction of the calculated gradient. When the algorithm 'converges' it reaches a local minimum which minimizes the error. There are different ways to propagate the error back in the network. One way is to calculate the error for the whole data set, another, calculate the error for each and every sample in the data set. However, calculate the error for small parts for the data set (a.k.a. mini-bathes) found to be work the best. Some issues of the gradient descent algorithm are caused by choosing an appropriate learning rate, it is not a straightforward process, and choosing it too small or big may cause the algorithm to stuck in bad local minima.

In order to solve some of the issues caused by choosing an appropriate learning rate and getting stuck in nonoptimal local minima gradient descent optimization algorithms are created. In this study, we used the A Method for Stochastic Optimization (ADAM)[38] optimizer which showed its use in many previous studies and learning tasks.

CHAPTER III

PREVIOUS WORK

Action recognition is a widely studied area that attracts interest from a wide range of fields including computer vision and robotics [39, 40], as it facilitates human behavior analysis [41] and human-robot interaction [42].

3.1 Action Recognition in Computer Vision

In computer vision, action recognition, in particular, pose estimation is a well-studied topic with approaches including monocular [43] or stereo-vision and depth camera-based point cloud approaches [44]. These approaches can either be based on manually designed feature matching with pose search or on gradient-based learning. For example, Keskin et al. [45] estimated hand positions with high accuracy using pose estimators that exploit multi-layered randomized decision forests.

Keeping this in mind, multiple studies of the first person (egocentric) perspective human action recognition studies have been conducted. In [46] authors used Convolutional Neural Network (CNN) [47] architecture to predict executed actions that are learned from hand motion cues. Ma et al. [48] also utilized a CNN architecture to learn scene and hand motion information. Garcia-Hernando et al. [49] used RGB-D images to train multiple state-of-the-art recurrent neural networks with a large corpus of video and mo-cap data to recognize different hand actions such as tearing, flipping and pouring.

Extensive research for action recognition in computer vision presented its usefulness for real-world scenarios. However, in these studies, the focus was not to be biologically relevant instead researchers widely prioritized to extract the most benefit from the current Machine Learning literature.

3.2 Action Recognition in Robotics

In some robotic applications, it may be possible to apply state estimation techniques to map observed behavior into a sequence of state descriptions compatible with the observer. This allows behavior understanding through estimating the value function of the demonstrator with the aid of adopting a reinforcement learning framework as exemplified by Takahashi et al. [50]. In general, in such scenarios, a range of methods from dynamic time warping (e.g. [51]) to inverse reinforcement learning can be applied (see [52]). In our work, we consider raw perceptual input and do not assume the existence of any state estimation capability.

3.3 Computational Mirror Neuron Models

Neuroscience research has shown that primate brains are endowed with multi-modal neurons (mirror neurons) that become active during the execution of hand actions as well as during the observation of similar actions when executed by a conspecific or experimenter [5, 6, 7]. It is generally accepted that the mirror neurons encode goal-directed actions and play a significant role in the understanding of observed actions of others' [53]. However, the underlying mechanisms and representations are still far from clear [54, 55]. Several mirror neuron models exist in the literature which may be considered as biologically realistic action recognition models.

As a biologically realistic action recognition model, an approximate brain model has been proposed by Oztop and Arbib [56]. They presented a system named Mirror Neuron System1 (MNS1) which models brain areas F5 (mirror neurons) and anterior intraparietal area (AIP) which are believed to play a crucial role in understanding possible grasping actions that can be realized. MNS1 approximates brain functions related to action execution and understanding. They achieve perspective invariance by defining a 'hand state' that describes the relation of hand with respect to an object, which may be considered as a special case (as it focuses only on grasping and not other actions) of the proposal pursued in this paper. Finally, an Artificial Neural Network replaces mirror neurons for learning and predicting actions.

Chaminade et al. [57] employed Hopfield [32] networks which are binary associative memory models with the ability to retrieve stored patterns. The underlying feature of the Hopfield networks is that they are working memory systems that are inspired by the human brain. In this context, from self-observations, their network learned and generated associations between visual input and motor commands. When subjected to visual input gathered from different hands, the system could generate correct motor commands to imitate the actions. Both of these systems learned from actions from self-execution and later used the experience they gained to make accurate predictions on understanding the actions.

Darwood and Loo [58] in their work they propose a computational model that is capable of imitation through self-exploration based on the associative learning capability of the Mirror Neuron System. Their model particularly focuses on the view-invariant facility of Mirror Neurons since they associate the motor acts with visual input. In their experiment in a simulation environment, they used a humanoid robot placed in front of a mirror and the robot was able to see outcomes of the motor acts that it was executed through a camera. Their network first learns to associate visual input to the motor commands. Later, their model is able to associate motor commands for observations from other perspectives.

Bonaiuto and Arbib [59] extending upon MNS1[56] based on the desirability metric they claim that the mirror neurons fire as to the result of actions which are not intentionally executed. Moreover, the attraction of action with signals from the mirror neurons may be learned through 'accidental success'. Additionally, reaching successful action execution is guided by the signal intensity. Therefore, attraction to an action and the probability of successfully executing it plays an important role in determining what to do next.

Nagai et al. [60] hypothesized that that mirror neurons system develops with the increasing resolution of visual input in the early development of an infant with regarding the evidence that behavioral and neuronal studies on early visual development presented. Their computational model trained with starting with low-resolution images of interactions with the robot and the resolution is increased through the process to imitate infant development. In the lower resolutions phase, the system interprets observed and self actions closely similar maybe as the same. Further through development (increasing resolution) self actions and actions executed by the others are better discriminated. In the end, maintained learned associations at early development create a basis for associating others' actions with motor commands and motor commands with self-observations. However, this work is limited to only hand movements.

Demiris and Johnson[61] inspired by neuroscience developed a system for action understanding and imitation for robotics. The architecture they come with is able to learn from both self-generated and observed composite actions in a dual manner.

Metta et al. [62] propose a mirror neuron system for action recognition based on the hypothesis that the mirror neurons system can emerge by the interaction of an agent with its environment. Specifically, their model learns to associate the motor states with the visual representation of the scene from the robot's perspective (self-exploration) considering the coupling between object affordances to the action type. Even though, their model successfully mimics the mirror neuron system it does not take a different view into account.

CHAPTER IV

SIMULATION ENVIRONMENT AND METHODS

With the goal of realizing an action recognition system based on the proposed action frame and assessing its feasibility for perspective invariant recognition, we designed a robotic simulation setup. The following sections describe this setup and the experiments conducted throughout this study.

4.1 Simulation Environment and Task Setup

We used the Gazebo[1] simulator [63] for the dynamic simulation of the TOROBO robot and Robot Operating System[2] (ROS) [64] (Kinetic) robot control architecture for communicating with the robot. TOROBO is an upper-body humanoid robot that has bi-manual manipulation capability with 22 degrees of freedom. In this study, we used only single-hand manipulation and considered a single-object, i.e. a vertically placed cylinder with a radius of 3cm and height of 13cm for beginning experiments, later we expanded our experiments to two more objects: a sphere and a horizontally placed cylinder that the robot can interact with. For modeling the case of action observation from different perspectives, we assumed that there are virtual observers around the table that might be observing the actor. The perception of those observers was either emulated by direct access to the simulator data (using an appropriate homogeneous transformation matrix to calculate observer viewpoints) or obtained by simulated depth cameras around the table (see Figure 5). The latter aimed at assessing the applicability of AF-based action recognition in physical robotic setups. Finally, to show the generalization ability of the Action Frame we repeat the experiments under

[1] `http://gazebosim.org/`
[2] `https://www.ros.org/`

two settings which are fixed and variable object positions. In the fixed object position setting the object is always placed at the same arbitrary position before the action starts, in the latter setting the object position is selected randomly in the robot's workspace.

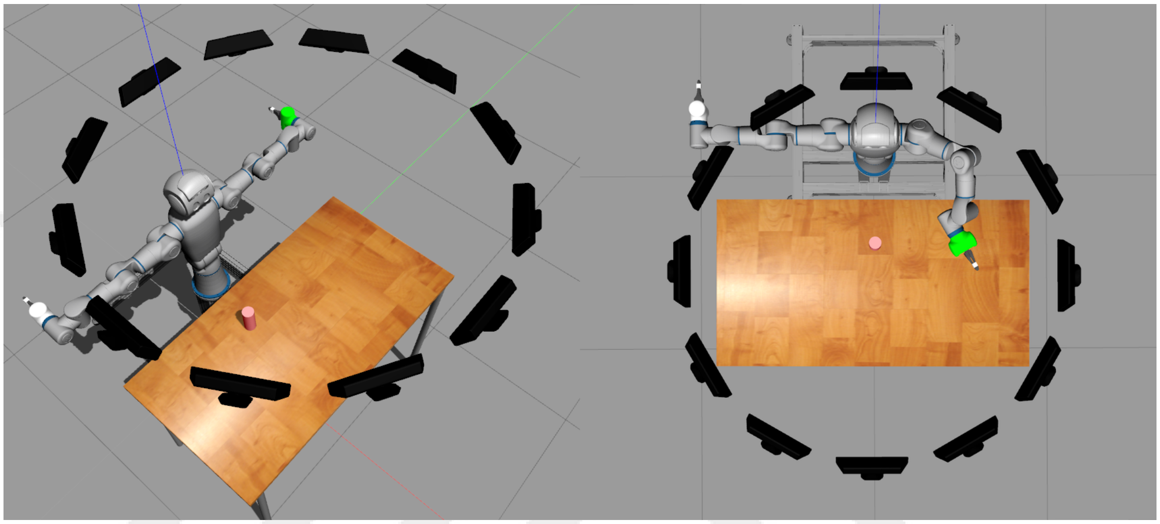

Figure 5: Table top simulation environment used is shown.

4.2 Action Repertoire of the Robot

4.2.1 Single-Object Experiments

To test the development of AF-based action recognition capability based on self-observation, we focused on four predefined parameterized actions. Out of the four, the two were simple actions of *push* and *poke*, and the other two were relatively more complex actions of *move-away* and *bring-to-mouth*. The trajectory and outcome of each action type were determined by two parameters: the angle of the robot gripper with respect to the object (action angle) and the location of the object prior to interaction. The execution trajectories of these actions were assumed to start from a fixed initial robot configuration.

Given the action type and its parameters, the joint trajectory to be followed by the robot is constructed and then communicated to the robot through ROS. First,

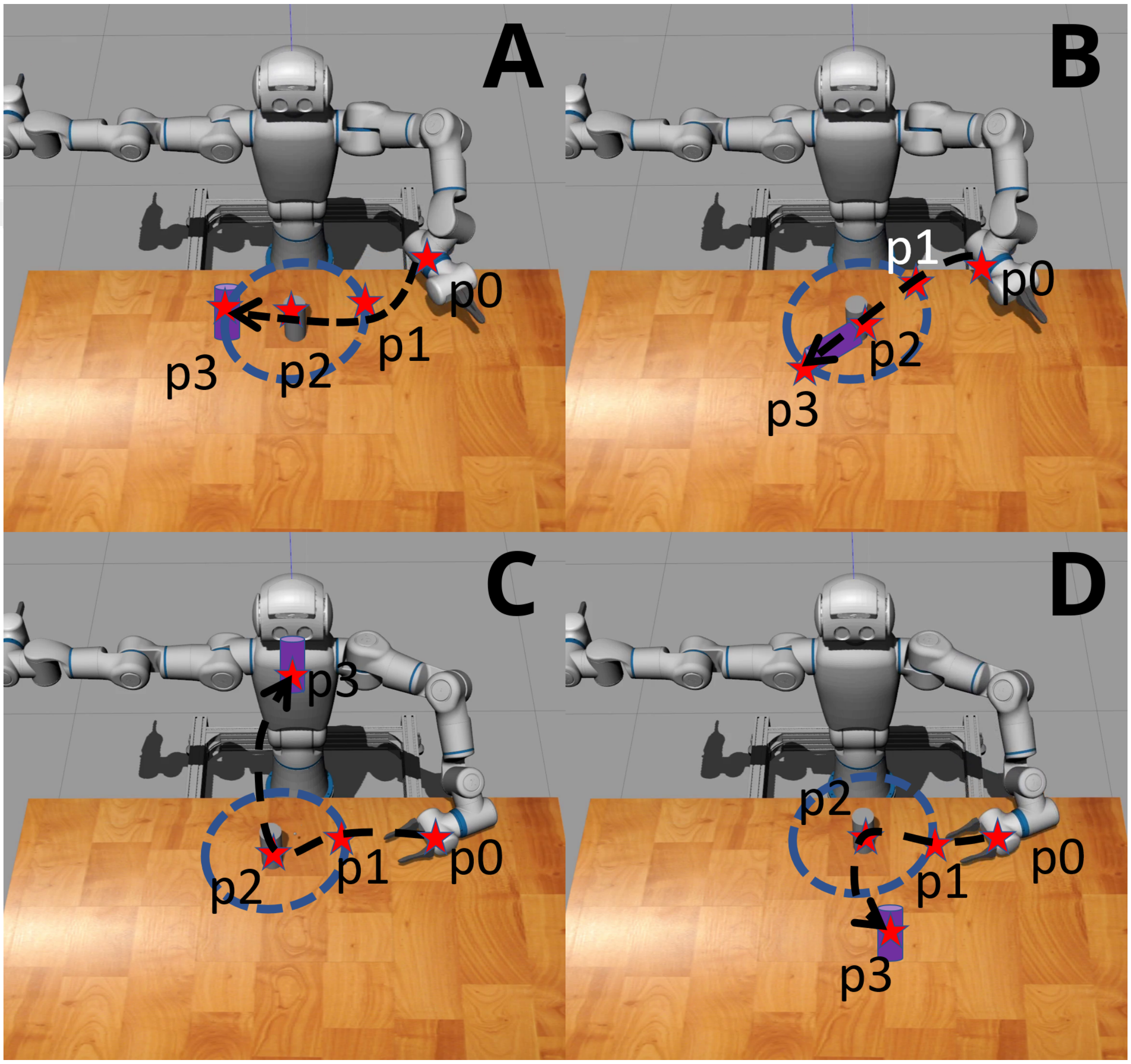

Figure 6: Action plan sketches is shown. (A:**Push**, B: **Poke**, C: **Bring-to-Mouth**, D: **Move-Away**)

four Cartesian space via-points are determined in an action-specific way, which are then converted into joint angles by using the inverse kinematics of the left arm and torso of the robot. Then the obtained set of joint angles are fitted with cubic splines to obtain smooth continuous curves for each joint of the robot. Finally, these curves are sampled at 20 equal intervals and fed to ROS to drive the robot arm and gripper. For all the actions considered, the robot is assumed to start its action at the same initial joint configuration determining the first via-point (P0); the second via point (P1) is determined by the angle parameter, which specifies a point on an imaginary circle (r=6cm) centered on the object. The next via-point (P2) is taken to be the center of the object, and the final via-point (P3) is determined according to the action type as described below.

4.2.1.1 ***Push and Poke***

For these simpler actions, the object is assumed to be at a fixed location in front of the robot (see Figure 6). The via-points for **push** and **poke** are formed by the initial gripper position (P0), and two symmetrical points (P1, P3) on the imagery circle centered at the midpoint of the object, which serves also as the middle via-point (P2). The only difference between the **push** and **poke** is that the vertical (z) coordinates of the P1, P2, and P3 are set differently. The z-coordinate is taken as 6.5cm for the **push** and 11.5cm for the **poke** actions. Since the object has a height of 13cm, the **push** action stably translates the object while the higher contact with the **poke** action often knocks over the object. While executing these actions the gripper is kept fully closed and the side of the gripper is used to form contact with the object. This facilitates a more robust contact and generates repeatable effects compared to using the tip of the gripper for establishing contact.

4.2.1.2 Move-Away and Bring-to-Mouth

For these complex actions, the goals of the actions are taken as specific locations that the object must be brought. In the case of **bring-to-mouth** action, the target of action is a fixed point in the proximity of the facial area of the robot, which determines the last via-point (P3). Likewise, in the case of **move-away** action the P3 is a fixed point which is away from the robot, near the boundary of the workspace of the robot (see Figure 9). As in the **push** and **poke** actions, the first via point (P1) is determined by the angle parameter of the **move-away** and **bring-to-mouth** actions. Similarly, the second via-point P2 is taken as the mid-point of the object. To enable grasping and transportation of the object to the desired target location, the gripper is commanded to enclose when the robot hand reaches P2.

4.2.2 Multi-Object Experiments

As described in the single-object experiments section a vertical sphere is used in the experiments. In multi-object experiments in addition to the vertical cylinder used, we added a sphere with a radius of 2.89cm and a horizontal cylinder with a radius of 2.89cm and a length of 13cm to better analyze the system capabilities against the different grasping variations and effects that may appear depending on the physical shape of an object. The vertical cylinder radius is also changed to 2.89cm. This radius is selected for the robot to make grasping easier of the sphere object kept in mind and also applied to other objects accordingly (see Figure 7).

In multi-object experiments, we discarded the **poke** action since the new objects are not suitable to tip over. Additionally, for **push** action, even though the push direction changes the horizontal cylinder is placed as its circle side looking towards the robot as displayed in the Figure 7. The intention is to observe the effects that may arise from the physical shape of an object. In this case, even though **push** action is applied from different directions, the cylinder generally starts rolling to the right from

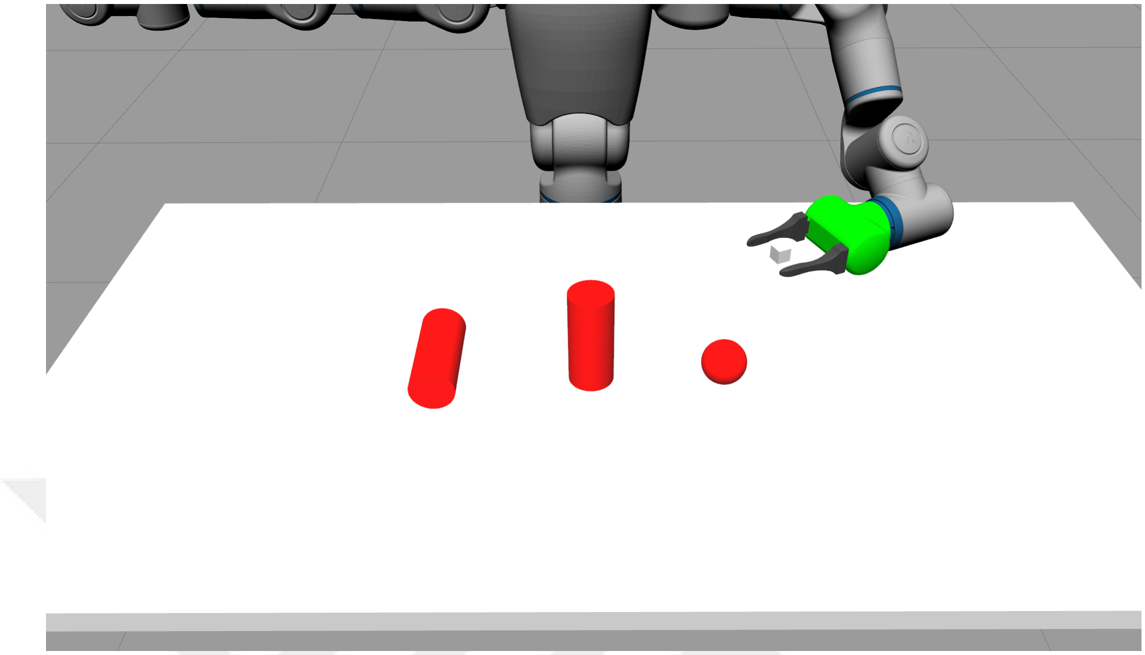

Figure 7: Showing the objects used in the experiments on a white table. Horizontal cylinder, vertical cylinder and sphere.

the robot's perspective and follows a direct path. Furthermore, for the actions which involve grasping, the horizontal cylinder is rotated with the action angle parameter to make grasping viable. These changes do not apply to the other objects since they have homogeneous surface areas.

In addition to the previous actions, we added two more grasping actions similar to the previous ones. These are **bring-left** and **bring-right** actions (see Figure 8) and they are defined similarly to **move-away** and **bring-to-mouth**, these actions again have a previously determined goal positions that the object must be brought. Moreover, action execution takes place the same way as before. However, in the multi-object case instead of assigning a random initial position for the object for grasp actions, this time all of the experiments (for all actions) separated into two categories as fixed and variable positions. The fixed position for the object in all experiments is an arbitrarily chosen point in the robot's workspace and locates directly in front of the robot. In the variable position case, as in the previous experiments, the point is sampled from inside of a circle in the work environment (on the table).

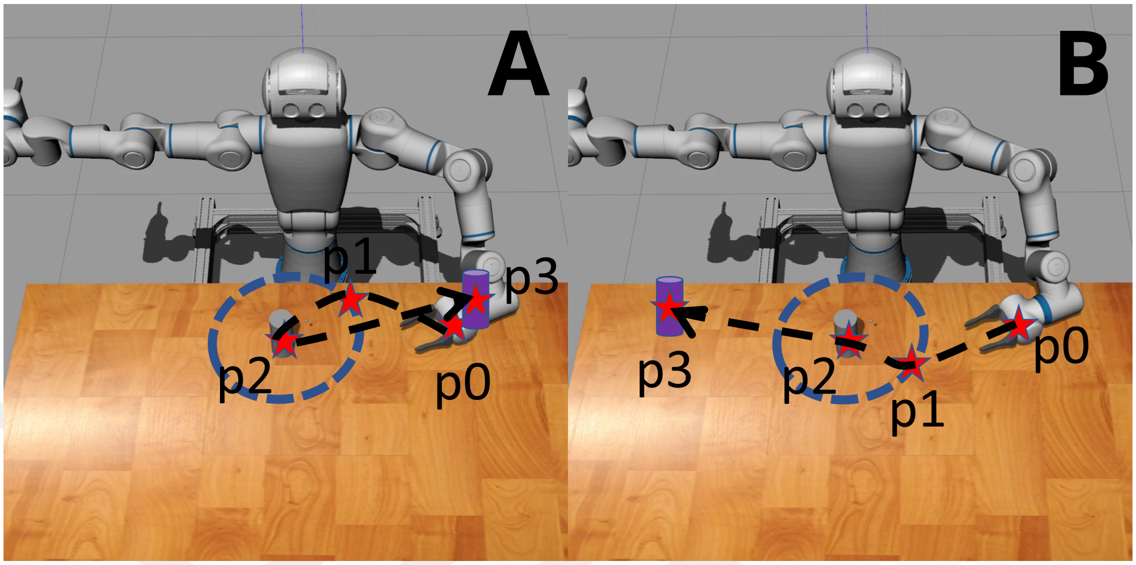

Figure 8: Action plan sketches is shown for newly added actions. (A: **Bring-Left**, B: **Bring-Right**)

4.3 Data Generation and Collection

4.3.1 Single-Object Experiments

As discussed in the previous section each action takes two parameters: action angle and position of the target object. For the Push and Poke actions, the action angle is sampled uniformly at random within a range of [-75, -105], and the object position is taken as a fixed position in front of the robot ([0.47, 0, 1.08]). The angle parameters for the **move-away** and **bring-to-mouth** actions are sampled within the range of [-35, -110], and the object position parameter is sampled uniformly from the interior of a circle (r = 7cm) that is parallel to the table and centered at [0.47, 0, 1.08] in world coordinates.

The simulated robot executed each action in 1000 different settings. While executing the actions, the robot is commanded through ROS and the desired joint angles are send to the robot with 10Hz interval (16Hz for **Push** and **Poke** actions). Data collection is carried out using the same interval, and the final object position is recorded after the auction ends. The arm and torso joint angles are also recorded to be able to recreate the results of the experiments later.

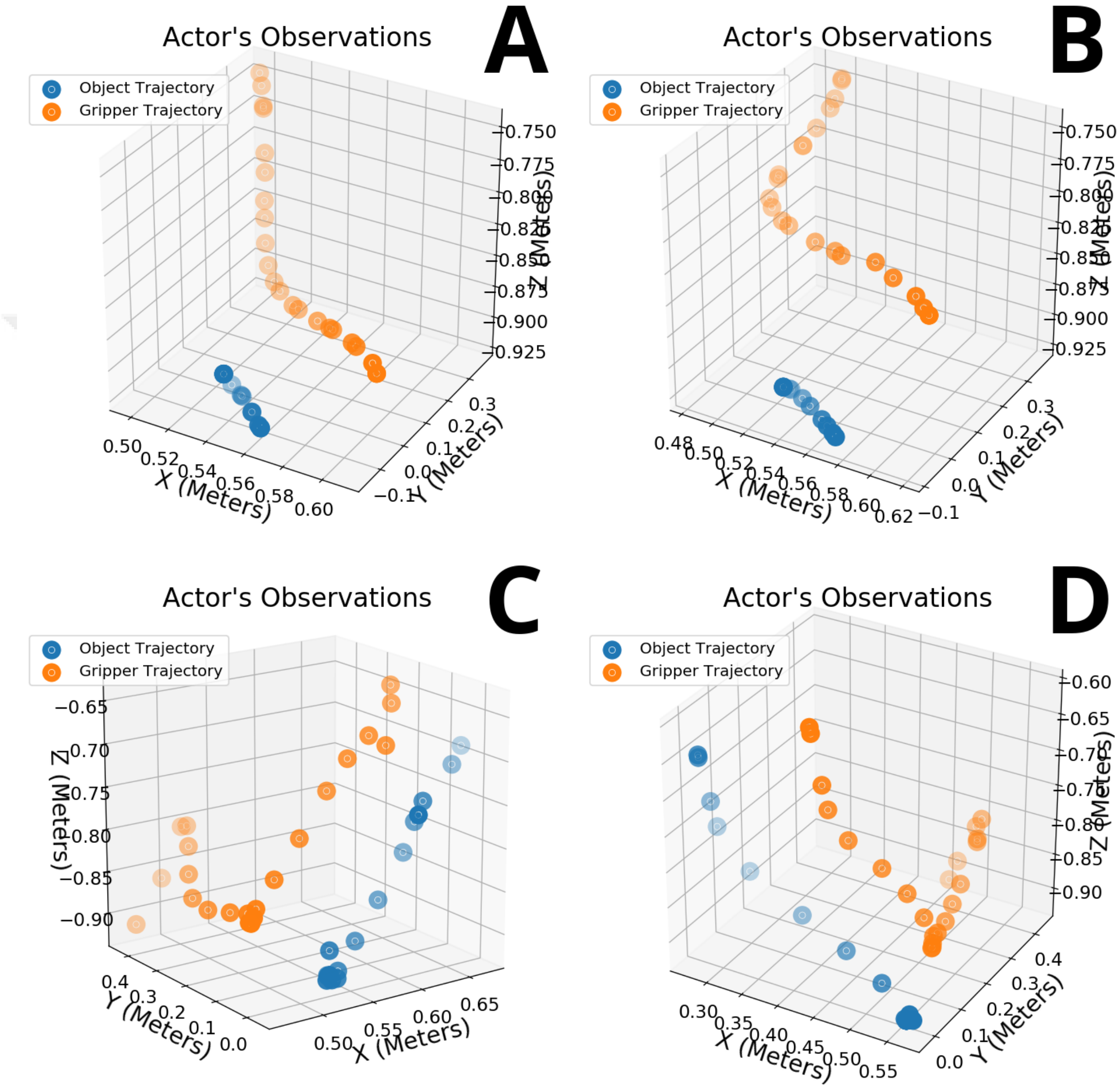

Figure 9: Sample object and gripper trajectories for each action. (A: **Push**, B: **Poke**, C: **Bring-to-Mouth**, D: **Move-Away**)

In order to overcome computational limitations in synchronized data collection, observer experience (i.e. positions with respect to the Egocentric Frame of each observer) are calculated after the action is completed by using the data of the actor via appropriate homogeneous coordinate transformations.

4.3.2 Multi-Object Experiments

In the multi-object experiments, most of the settings used and action parameter ranges are kept the same in the single-object case, this includes the newly added actions **bring-left** and **bring-right** (see Figure 10). As mentioned in the previous section, multi-object experiments consist of two categories: fixed and variable position experiments. In the fixed position experiments, the current object is placed always at [0.516, 0, 1.08] on the table and directly in front of the robot. In the variable position experiments, the current object is placed at a position that is randomly sampled from inside of a circle with a radius of 7cm centered at [0.47, 0, 1.08] similarly in the single-object case since it has been proven to be a viable workspace for the robot.

Furthermore, the gripper angle is changed to 160 degrees for grasp actions when the object in question is the sphere or horizontal cylinder. In other words, the gripper angles are changed to have the tip of the gripper looking downward towards the table. This eliminates the possibility of collisions that may occur between the gripper and the table during actions since object heights are small and close to the table. However, the gripper angle is kept the same for the vertical cylinder as in previous experiments (single-object) at 90 degrees.

4.4 Object and Hand Localization via Depth Camera

Since we plan to deploy the developed perspective invariant action recognition system in the real world and explore its possible use cases in robotics as a functional mirror neuron architecture, we designed a simple color-based perception system so that the robot can 'see' and track the target object and the hand in action. This way, it would be easier to transfer learning and prediction to the real world. During action execution in the simulator, we also performed data collection of hand and object positions by using the emulated Kinect depth cameras. In order to gather this data, the depth camera outputs are processed with the help of the OpenCV library [65]. For

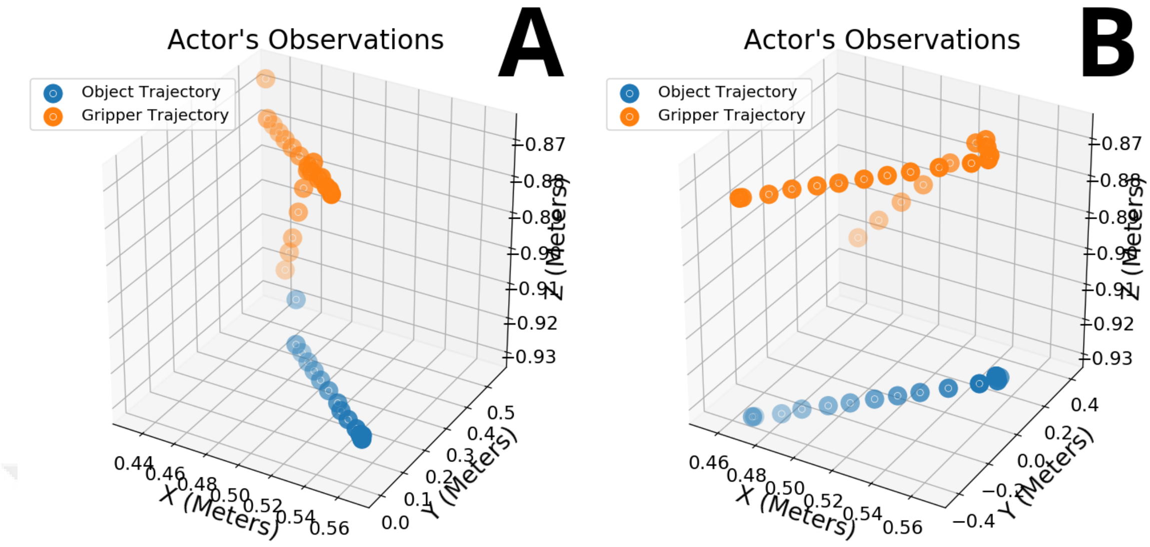

Figure 10: Sample object and gripper trajectories for new actions with vertical cylinder object. (A: **Bring-Left**, B: **Bring-Right**)

computational convenience, the processing for object detection is based on 3D color segmentation. Consequently, the object and the gripper are given distinct colors, red and green, respectively. Given an image frame for color, filtering is applied using the 'inRange' method of OpenCV which gives a set of points for each color. The centers of the extracted point clouds are then found by using the 'findContours' and 'moments' methods.

In order to accelerate the computations, both image and point cloud data are down-sampled by a factor of 8. One limitation of using object detection in this fashion is that view-dependent occlusions may create offsets in the point-cloud centers corresponding to the gripper and the target object.

Finally, since multi-object experiments cameras rotated to look downward to the table in the simulation environments, after the action execution and the location data is collected from the depth cameras the error caused by the orientation of the cameras is corrected computationally with a rotation matrix. Also, the shades are disabled in the simulator to increase object detectability.

4.5 Action Coordinate Frame (AF) Construction

The AF is constructed based on the Gravity vector (g) and the velocity vector (v) pertaining to a hand in action. The velocity vector is in general a function of time over the action period. In this study, we take the velocity vector of the hand at the moment when it enters the vicinity, i.e. 20cm proximity of the target object. The velocity vector is estimated by numerical differentiation, and its projection to the horizontal plane (v_{proj}) is used for setting up the Action Reference Frame (AF). Figure 11 shows the AF overlaid on the initial position of the object. In detail, AF is calculated as follows:

$$x_{axis} = v_{proj}/||v_{proj}||$$

$$z_{axis} = -g = [0, 0, 1]$$

$$y_{axis} = x_{axis} \times z_{axis}$$

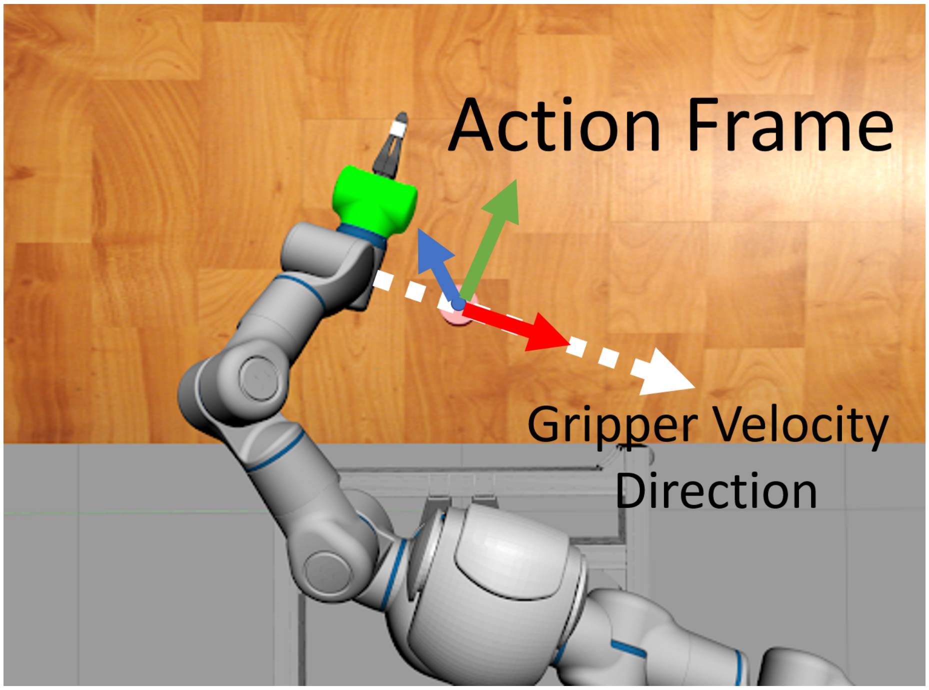

Figure 11: Illustration of the Action Frame (AF). Gravity and hand velocity vectors are used to construct the AF (red:x-axis; green:y-axis; blue:z-axis).

4.6 Predictive Learning Network

To model a predictive system that can be trained by self-observation, we formalized the problem as learning to predict the action-code, action-parameters and the effects given an object and hand in action.

4.6.1 Single-Object Experiments

In particular, for single-object experiments, the input for the predictive system is taken as 3D hand positions of 5 consecutive frames represented in either EF or AF. The output, on the other hand, corresponds to the effect that would be generated, the action type and action parameters corresponding to the action being observed. Thus, the size of input to the neural network is 15 (5 positions) and the size of the output size is 11 (the effect encoded as a 3D offset vector (3), the one-hot action code (4), and the action parameters encoded with the action angle (1) and the initial position of the object (3)). The system starts storing hand positions after the hand approaches to the vicinity of the target object (enters within 20cm range of the object), and after 5 observations are done, the system produces its prediction.

With this input-output specification the prediction system is implemented with a three layer fully connected Artificial Neural Network (ANN) with 16 neurons in each layer. We used *rectified linear units* (Relu) [36] for the network activations. The size of the network is empirically tuned to be small yet capable of learning the prediction problem targeted.

The training and testing data is scaled between 0 and 1 using a min-max scaler. For every experiment the network is trained 4000 epochs with learning rate 1e-3, batch size of 64 and the same random seed is used in training to exclude randomness. Finally, A Method for Stochastic Optimization (ADAM)[38] optimizer is used. No regularization technique is used since the networks could be considered shallow.

4.6.2 Multi-Object Experiments

In Multi-object experiments network input is extended with one-hot object type code therefore the input vector size increased to 18. Additionally, the output is increased with the addition of new actions to 12 (for network structure see Figure 12). Most of the settings used in the initial experiments are kept the same such as hidden layer neurons size with 16 neurons, learning rate, batch size, and so on. One change is however made to the training epoch and increased to 5000. Finally, the ten networks that are used in the experiments are initialized with the [7, 13, 26, 32, 48, 56, 5, 357, 9, 845] random seeds.

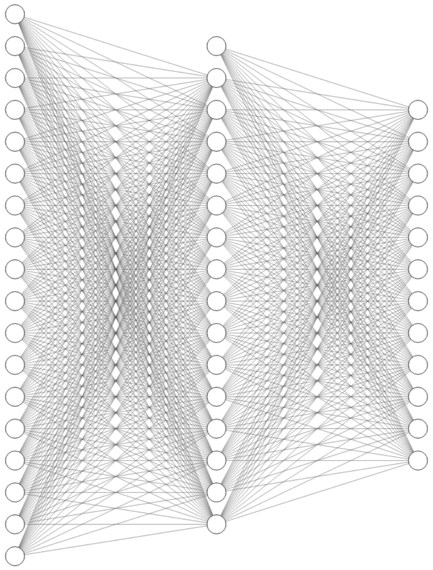

Figure 12: Illustrating Neural Network for Multi-Object Experiments.

4.7 Experiments conducted

4.7.1 Single-Object Experiments

By using self-observation data, we trained two separate networks: one that represents the data in the Action Frame (AF), and one that uses the Egocentric Frame (EF) representation. After we ensured that the predictions with self-collected data and both representations are successful Subsection 5.1.1, we switched our attention to contrast the capabilities of AF- and EF-based predictions when they are used to make predictions of others' actions Subsection 5.1.2. Note that, in general, an egocentric reference frame has an origin aligned with the observing agent. However, training a system with self-observation and then attempting to do prediction based on the observation of others would create large offsets in hand positions due to the simple fact that one's own hand is often much closer than others' hands. Therefore, to improve the prediction capability of EF-based prediction, we translated the origin of EF to the object center, as we did for the case of AF.

Every action is sampled 1000 times as described in the Data Generation and Collection section. We used an 80/20 train-test split on the data.

The training set includes 800 randomly selected samples from 1000 samples generated during simulation for every action, therefore the size of the training set is 3200. The test set has 2400 samples gathered from each observer and the actor (12 observers in total). The network prediction error is calculated using the *mean squared error* (MSE) loss function.

Finally, to test the performance of both EF- and AF-based predictions of others' actions, we repeated the latter experiment by using the emulated depth camera image.

4.7.2 Multi-Object Experiments

The multi-object experiments are conducted with the same general structure as the previous experiments. The difference is that since the multi-object experiments have more combinations, the data set size increased significantly. As keeping the same train-test split with three objects the training and test set size grew to 12000 and 3000 for each observer (36000 in total) respectively. However, for these experiments, the first 200 samples were used as the test set and the rest is used for the train set as opposed to single-object experiments.

The experiments were conducted under two different settings as fixed and variable object positions with K-Fold cross-validation with 10 networks, 40 neural networks are trained in total. The results are presented in Chapter 6.

CHAPTER V

SINGLE-OBJECT EXPERIMENTS

In this section, we present the predictive learning results based on self-observation learning by using representations in Egocentric and Action Frames. Then we present the results showing the generalization ability induced by these reference frames for observation actions of others. Finally, towards a real-world implementation, we present the learning and generalization results based on depth camera-based object and hand perception.

5.1 Experiments with Simulator Provided Location Data

5.1.1 AF and EF based learning of self-generated data

Figure 13 shows the RMSE error for predicted action angle and Euclidean distances of effect throughout the training process for both AF and EF networks. It is evident that both networks show a convergent learning regime with the loss approximately stabilizing towards the final epochs. So we can deduce that the networks designed are suitable for learning the data derived from our setup.

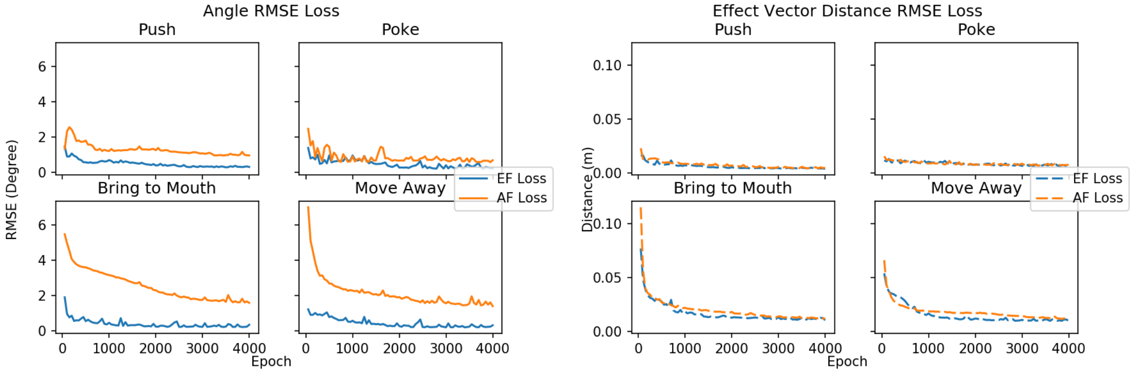

Figure 13: Test RMSE loss for action angle and effect distance for both networks. AF is Action Frame and EF is Egocentric Frame losses.

5.1.2 Observing Others via AF- vs EF-based Prediction System

After training is completed on self-observations, both AF and EF networks are tested on previously unseen data perceived by the observers. Each observer perceives the world through its eyes (or cameras), thus the eye-centered/egocentric representation is dependent on the pose of the observer. Note that in the stage we are considering, each observer can only learn from their own actions. Therefore, a change in the pose of the observer considerably affects the prediction capabilities of the observer's prediction if it is based on an Egocentric representation. For understanding the actions of others, additional mechanisms or different representations seem necessary. We took the latter alternative and proposed the Action Frame. When the actions are seen through the Action Frame, what the observer and the actor 'see' is very similar, and indeed in a noise-free simulation environment, it is identical. In the real world, there would be perceptual noise, occlusions, and distortions that would create imperfections. The results in this section show these arguments quantitatively.

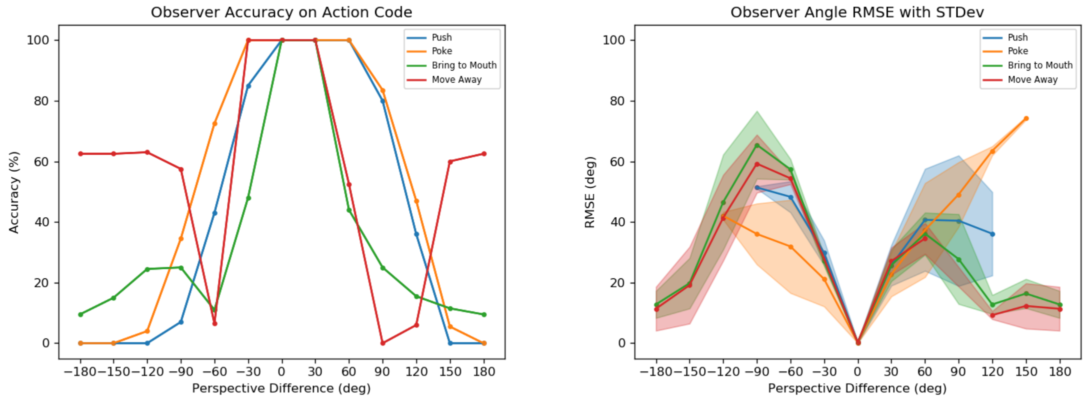

Figure 14: Action understanding performance with EF based representations by using noise-free data. Left: recognition accuracy as a function of viewpoint difference. Right: Action parameter (angle) prediction accuracy for those actions recognized correctly. The shades indicate standard deviation.

The leftmost plot in Figure 14 shows the action prediction accuracy of the network trained with data using the EF representation. As can be seen in the graph, the more the observer wanders away from the viewpoint of the actor (i.e.position around the table), the worse the prediction accuracy gets. This is outcome expected since the observer has no experience related to the other viewpoints. Still, the generalization capability of the neural network generates somewhat correct predictions related to the observed action for neighbor observers (i.e. viewpoints); but, for most of the actions, the accuracy goes to zero when the observer is for example, directly opposite from the actor. In the same Figure, the rightmost graphs show the action angle prediction error for each action as the root-mean-square error (RMSE) with standard deviation. Similar to the action recognition performance, the further the observer wanders away from the actor's perspective, the more the loss increases. Note that, for error calculation, only the action samples that were correctly predicted were used. The low angle errors and standard deviation at viewpoints where action recognition error is high indicate that for those observers only a few actions can be recognized but when they are recognized their action parameters could be reliably retrieved.

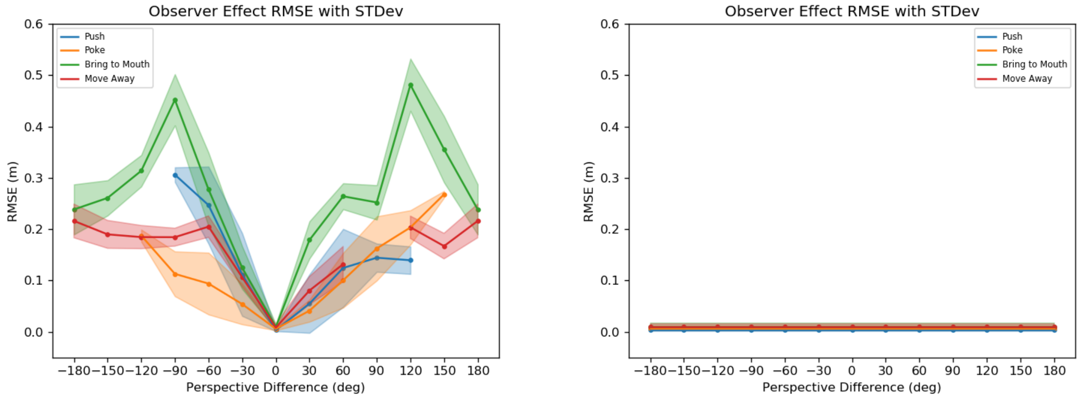

Figure 15: Effect prediction performances with EF and AF based representations. Left: accuracy based on EF, Right: accuracy based on AF as a function of observer viewpoint difference. The shades indicate standard deviation.

When the network is trained with AF representations, we get perfect prediction accuracy since the network is trained with noise-free data and thus the observer experience is the same as the actor in AF representation. Similarly, the action angle error is close to zero. A similar contrast between AF and EF effect prediction can be seen in Figure 15.

5.2 Experiments with Depth Camera based Location Data

5.2.1 Observing Others via AF- vs EF-based Prediction System using Depth Camera Input

We conducted the same experiments as the previous section by using object and hand position data obtained via the Kinect-based perception system. Additionally, we used splines to replace the missing data points caused by the occlusions. However, this procedure is not employed in the following experiments since it is concluded that it actually decreases prediction accuracy (see Chapter 6). Figures 16 and 17 show action recognition performance for EF and AF based learning respectively. As expected, the networks trained with the emulated Kinect data show poorer performance compared to the results obtained by using noise-free data from the simulator.

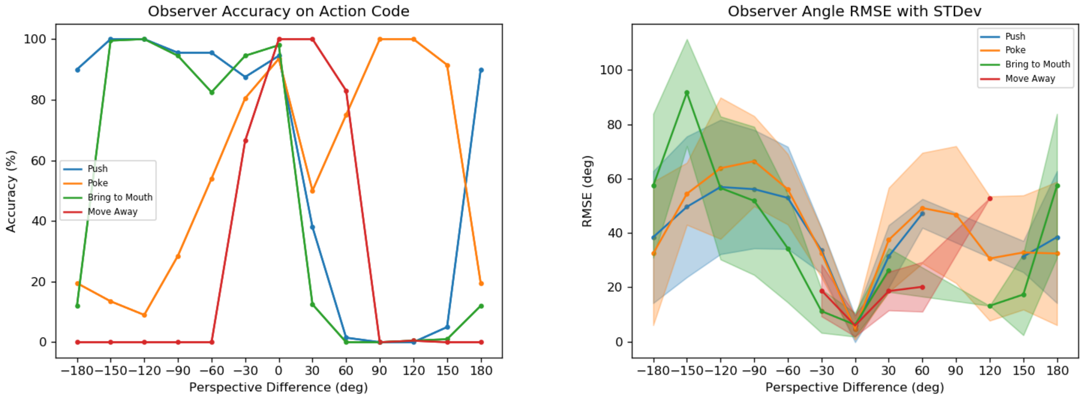

Figure 16: Action understanding performance with AF based representations for emulated Kinect data. Left: recognition accuracy as a function of observer viewpoint difference. Right: action parameter (angle) prediction accuracy for those actions recognized correctly. The shades indicate standard deviation.

Even though depth-sensing is itself a simulation, there are certain perceptual biases and occlusions depending on the viewpoint. The results from our experiments suggest that with non-perfect perception still a high level of perspective invariant action recognition capability can be obtained if we use AF-based representations. This observation is also valid for effect prediction as shown in Figure 18).

It is worth noting that the asymmetric performance drop seen in Figure 16, when compared with the symmetric performance drop of Figure 14, indicates that the imperfection in the implemented hand and object perception system manifest itself differently for each action.

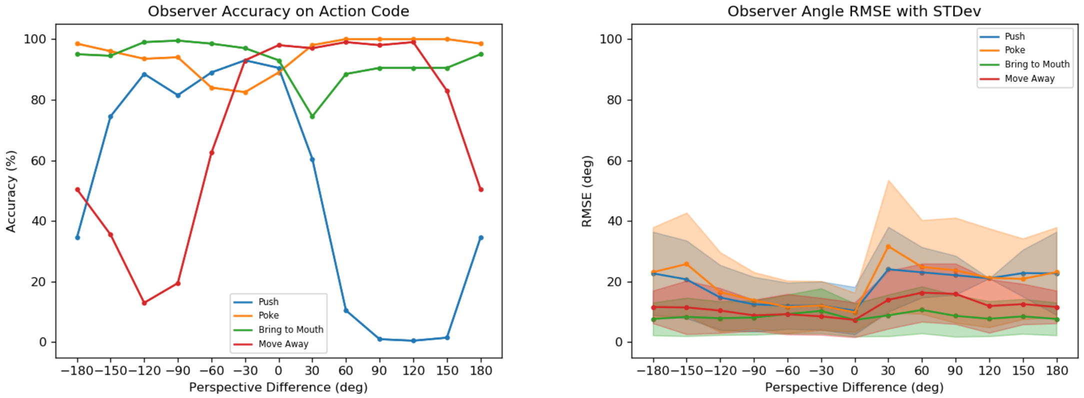

Figure 17: Action understanding performance with EF based representations for emulated Kinect data. Left: recognition accuracy as a function of observer viewpoint difference. Right: action parameter (angle) prediction accuracy for those actions recognized correctly. The shades indicate standard deviation.

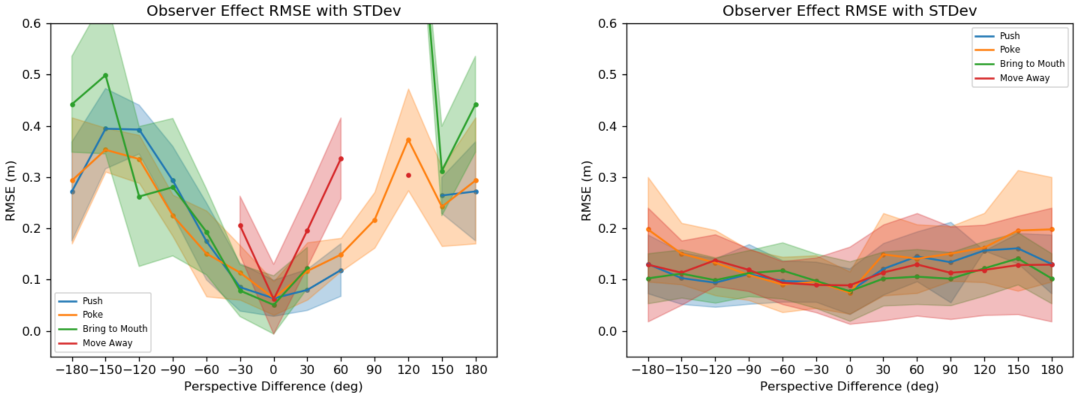

Figure 18: Effect prediction performances with EF and AF based representations for emulated Kinect data. Left: accuracy with EF, Right: accuracy with AF as a function of viewpoint difference. The shades indicate standard deviation.

CHAPTER VI

MULTI-OBJECT EXPERIMENTS

In this section similar to the previous section we present the results of self-observation learning by using representations in Egocentric and Action Frame on multiple objects (sphere, vertical cylinder, horizontal cylinder) under two settings, namely fixed and variable objects position settings, and show how the system generalizes using representations on EF and AF. The setting determine the object location before the action starts as described in the Chapter 4. We also present learning and generalization results based on depth camera based object and hand perception into a step for real world application. Finally, we analyze the limitations and performance of the depth perception system. In the Figures 19 to 24, also in 26 and 27 attached at the appendix, action accuracy, predicted angle and effect root mean square errors (RMSE) can be seen respectively as three rows, columns correspond to object type (sphere, vertical cylinder, horizontal cylinder), detailed examination is presented in the later sections. Each graph shows the mean and standard deviation of ten different trials in which the networks are initialized with previously selected different random seeds. One point that should be noted is that while accuracy graphs show the mean and standard deviation values directly, in the angle and effect prediction error graphs as previously, prediction errors for only the correctly predicted samples are taken and results are produced by taking the average over ten networks' results. To be concrete, for every point on the graphs (all observers) average of the prediction error and the standard deviation is calculated and presented in the prediction error graphs.

6.1 Experiments with Simulator Provided Location Data

6.1.1 Fixed Object Position Setting

In Figure 19, Egocentric Frame network output can be seen with the previously described structure. When the action code prediction accuracy is examined one common item that can be seen is that **push** action accuracy seems conspicuously high even though the network is trained with Egocentric Frame representations and tested on unseen data from different perspectives (translated coordinates). Also compared to the other actions standard deviation is much smaller. The cause of these observed results could be due to the simple nature of the **push** action in addition to the pronounced difference of the gripper trajectory compared to the other actions. Under these circumstances, a relatively small neural network can correctly classify the action code. Another thing to consider is that since one-hot encoding is facilitated the maximum valued index is accepted as the class id, this can lead to high classification accuracy even when the network confidence is low, although this is not explicitly checked and confirmed in this case.

When we consider the other actions, the results create a bell shape, as expected the action code prediction accuracy decreases when the perspective difference in degrees increases (moving away from the actor's perspective) and at 0 degrees (actor's perspective) since the network has converged successfully the action code correctly predicted for every sample in the test set. However, results show a high standard deviation, especially with the increasing perspective difference. This is because a minority of the networks are able to correctly identify the action code for observers even though they are away from the actor's view that is why the standard deviation has an increasing trend while the perspective difference degree increases. This trend is especially evident for **bring-right** action for the objects sphere and vertical cylinder. We believe that the low complexity of the feature space plays a part in these results. To be concrete, under the current circumstances actions which include grasping starts

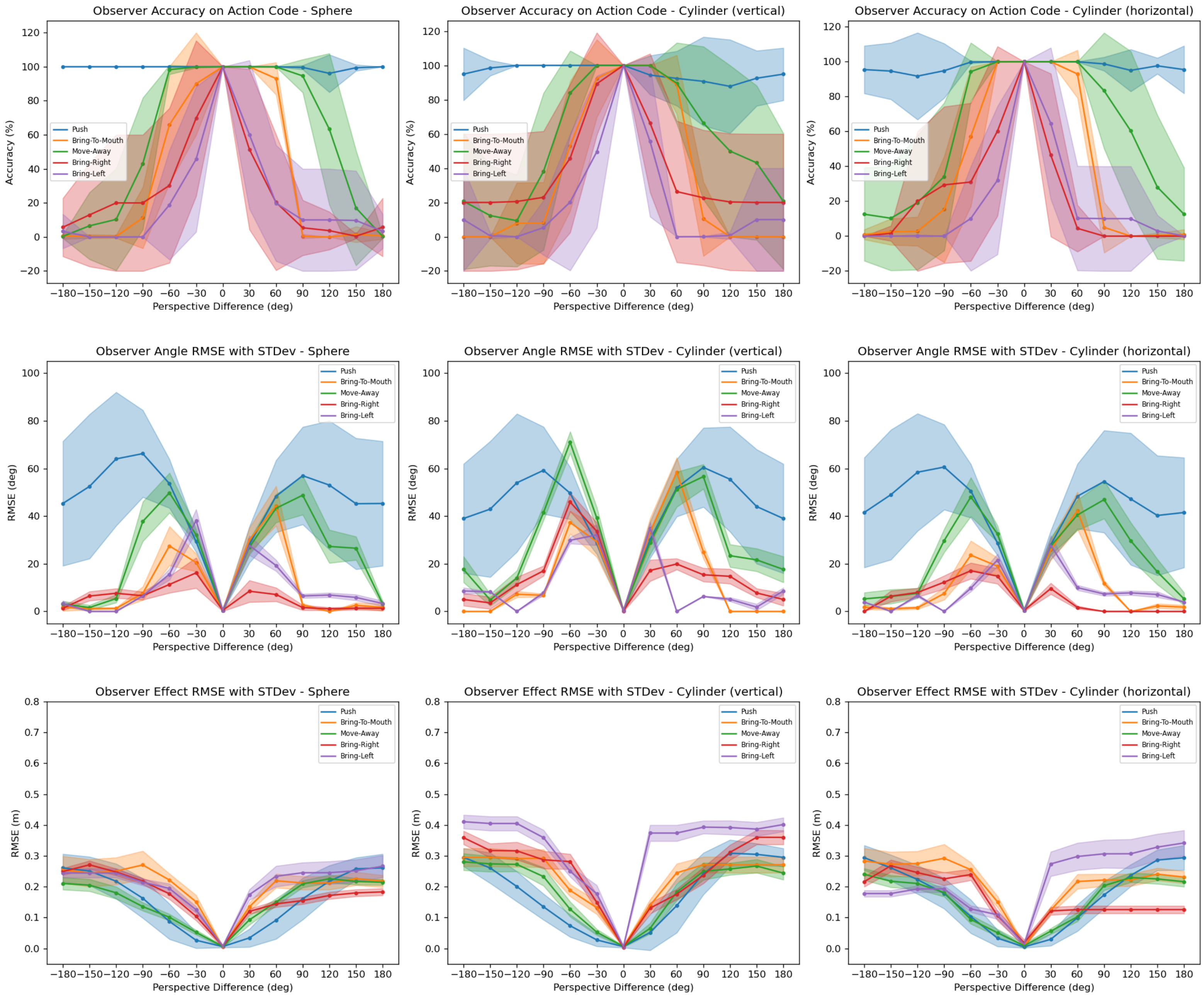

Figure 19: The Egocentric Frame under Fixed Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for simulator provided location data.

and ends at the predetermined locations. Even though the action angle parameter changes the gripper trajectory cubic spline that is used at the action execution step takes its shape according to the end location and indicates which action is taking place. This lowers feature space to changing only a single parameter and depending on the network initialization some times the neural network is able to converge to a point that is successful to classify action codes even though input to the network from a different perspective. One additional note is that clean simulator data is also taking a part in this case since action space is sparse and relatively easy to solve/classify.

With the change of perspective difference, the more an observer moves further away from the actor's perspective the more error increases as expected from the representations in EF. However, we see the error goes down with the increase in perspective change, that is because graphs display the error for only test samples with their action codes predicted correctly, that is why after around 90 degrees the error goes back down and the network is able to predict the action angle correctly and for the small sample group whose action code predicted correctly, the networks predict with high confidence. The **push** action being the only case differentiating from the others. We again see similar behavior to angle prediction of other actions with the decreasing prediction error even though the action code accuracy is high for **push** action. However, this time the networks' confidence is low indicated by the high standard deviation.

In the third row in Figure 19, we see a similar result with angle prediction error for the effect prediction error, at the center (actor's view) the error is almost zero, with the perspective change the error gradually increases. One striking difference at the first glance is that instead of going down again the error converges to some point. Additionally, the **push** action effect error curve follows a similar path to what other actions' errors draw. This behavior may be caused by the higher dimensions of the feature space of the effect compared to the action angle parameter since effects are denoted by 3D position change vectors whereas the action parameters are scalar.

In Figure 26 attached at the appendix, we can observe the result for action frame for fixed object position. Under the current setup (with clean simulator data), when the network is able to converge to a minimum (trained) since representations in AF are the same for all observers the networks' prediction results show minimum error for all observers. Moreover, we see that for all actions action prediction accuracy is %100. Consequently, action angle parameter and effect prediction errors are minimal. That is to say that all observers perceive the action as if they are executing the action and since all observers are accepted as they have the self-learned experience of action execution this invokes mirror neuron functionality by using the Action Frame and a neural network (creating the same state in their brain/network).

Finally, we do not observe any significant changes in the results for different object types under the current setup.

6.1.2 Variable Object Position Setting

One variation added to the multi-object experiments is variable object position described in Chapter 4. In Figure 20, we see the result of these experiments with variable object positions. Compared to the fixed object position experiments, the action code accuracy declines more steeply and it is evident that the standard deviation is higher. Expectedly, since the Egocentric Frame is also placed on the object (but with the observer's orientation who is perceiving the action) it performs similar to fixed object position experiments. We observe the increasing standard deviation effect also in the angle parameter prediction error, however, it is not a dramatic increase. As in the case of effect prediction error, it seems to be the least affected by the variable object position. The most prominent change is observed in vertical cylinder's **bring-left** and **bring-right** actions.

In Figure 27 attached at the appendix, results of variable object position setting

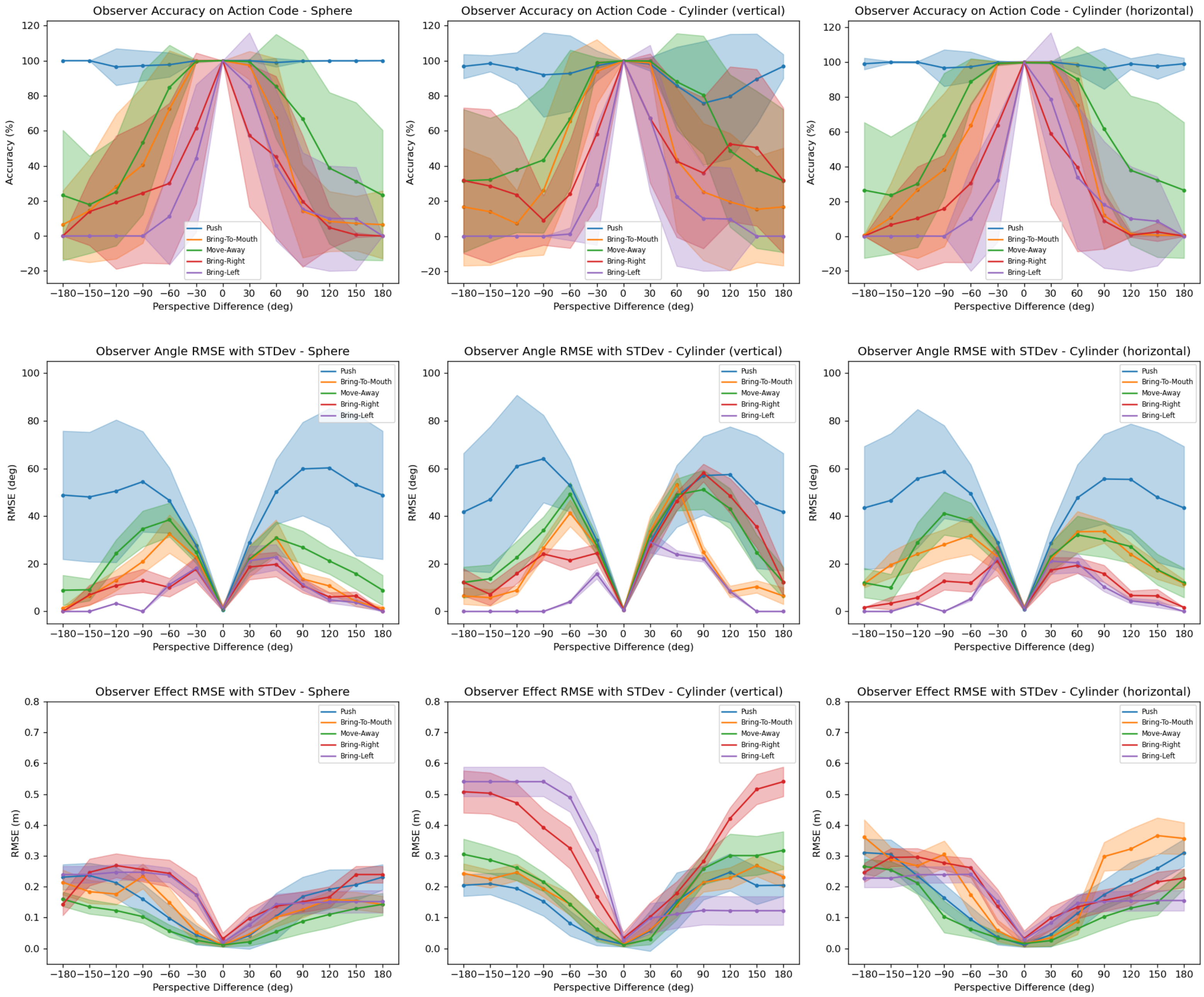

Figure 20: The Egocentric Frame under Variable Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for simulator provided location data.

experiment is presented. The results show a high correlation with its counterpart (the fixed object positiın setting) action accuracy showing the same performance and the only difference being the small increases in the prediction errors.

6.2 Experiments with Depth Camera based Location Data

The same experiment data is also gathered thru Microsoft Kinect cameras placed into the simulation environment to be able to move the simulation setup to the real world and robot later. In this section, we present the results of the same experiments carried on the data which the Kinect cameras gathered and discuss the effects/limitations of realizing such a system.

6.2.1 Fixed Object Position Setting

In Figure 21, compared to the Figures 19 and 20 we see the most prominent changes in **push** action accuracy and overall standard deviation on all of the action accuracy performance of the networks. Obtaining the current object and the gripper positions thru Kinect cameras adds a small error since our object detection system gives out the object position as the closest point to the depth sensor which the camera is able to see. This error added on top of the difference created by changing the perspective results for **push** action accuracy to follow a more similar path to other action accuracies and therefore behave more naturally and expected. The second thing to consider is occlusion that may happen throughout the action execution, this is more explicit for the smaller/shorter objects such as sphere and horizontal cylinder. Prominently, the observer at -30 degrees suffers from these occlusions.

The action angle prediction error differs from the previous examples especially for **push** action, we again see closer behavior to other actions similar to the action accuracy results. However, the most drastic changes are the error spikes at -30 degrees. These are caused by the occlusions (the camera is unable to detect the object in the scene). This issue does not arise in the vertical cylinder case since it is taller than

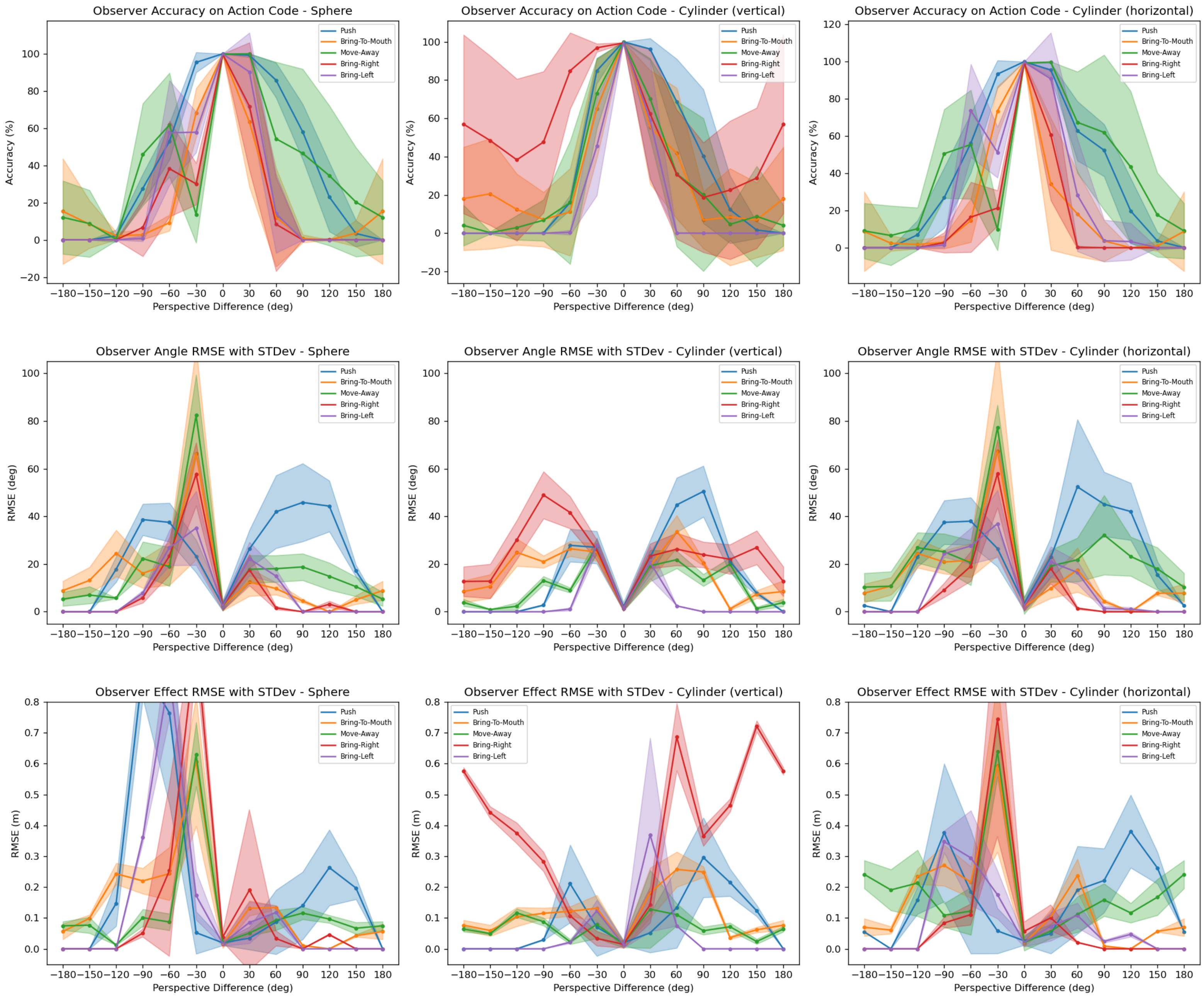

Figure 21: The Egocentric Frame under Fixed Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for depth camera based location data.

other objects and can be detected even if the object is behind the arm or the gripper.

The effect prediction error unlike the other results is not consistent among the three object types. For example, the error spikes originated from the occlusions still present for sphere and horizontal cylinder objects. Additionally, for **push** and **bring-left** actions of sphere displays similar error spikes for -90 and -60 degrees respectively with no trace of occlusions took place in other graphs for these particular observer positions. This may be caused from the small size of the sphere object As in the case of vertical cylinder high accuracy of **bring-right** action reflects as an upside-down bell shape at the effect prediction error as expected. Lastly, other than the error spikes on the error at -30 degrees, horizontal cylinder results fall parallel to its action accuracy and the angle prediction error results.

In Figure 22, compared to its counterpart (experiment with simulator provided data) how the error produced by the object detection system is affecting the predictions can be better observed in the middle graph (vertical cylinder) since the object is always seen by the cameras and not subjected to the occlusions thanks to its size. Moreover, performance issues discussed above and caused by the occlusions are better identified with sphere and horizontal cylinder results (additional issues related to the object detection system will be disclosed in Section 6.2.3). For these two objects, we see a significant drop in the accuracy and an increase in prediction error at -30 degrees throughout the results. Another common drop is seen at 60 degrees, interestingly it seems not to be caused by the occlusions but rather related to the other issues which will be discussed later.

Furthermore, even though the prediction error is increased compared to the previous experiment with the simulator provided data, compared to the Egocentric Frame observers which are equipped with the Action Frame can successfully understand the action and its possible effects on the environment.

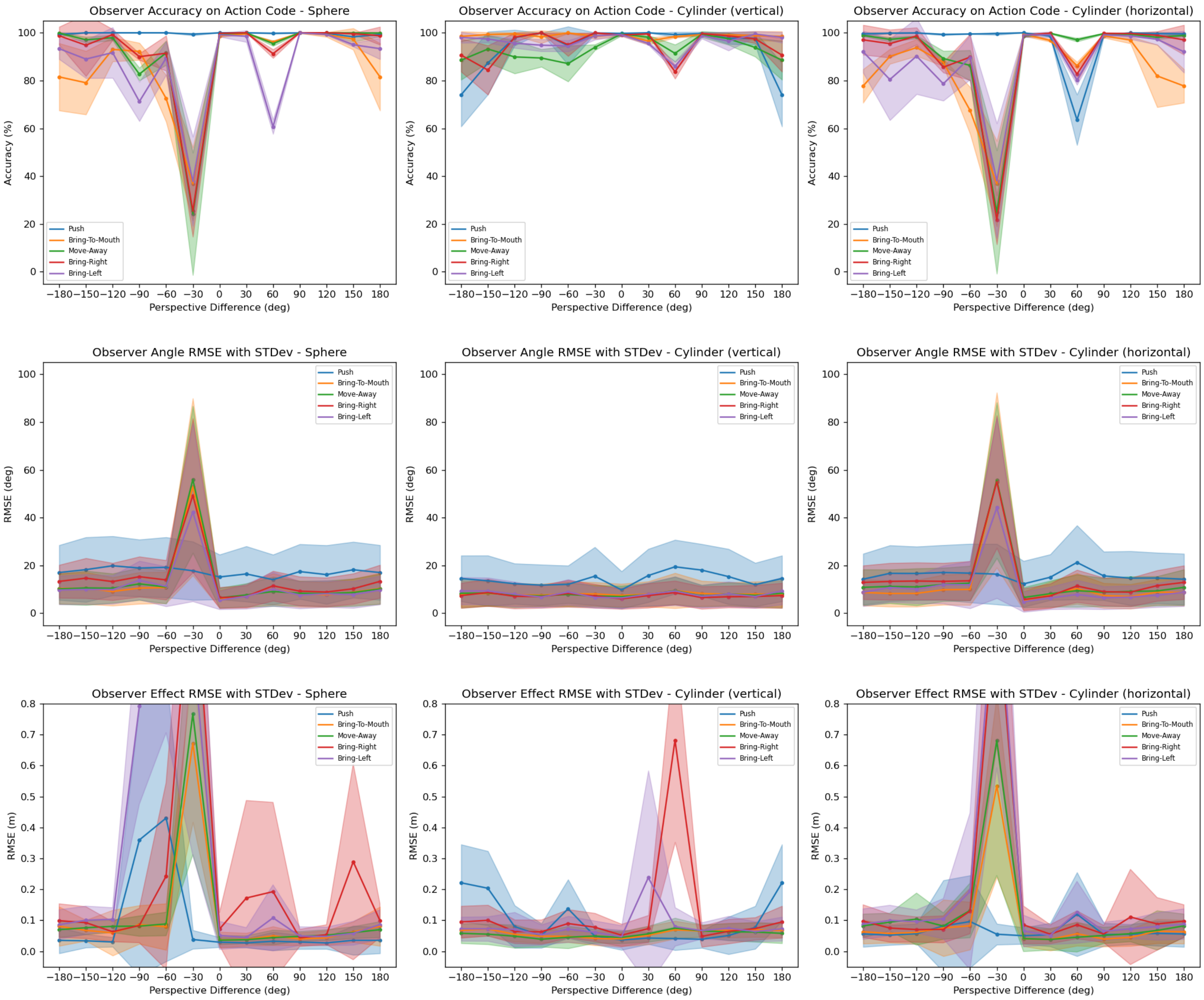

Figure 22: The Action Frame under Fixed Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for depth camera based location data.

6.2.2 Variable Object Position Setting

In Figure 23, we see similar results to fixed object position setting experiment. The subtle differences are as follows, action accuracies are improved ever so slightly in the variable object position setting in addition to having wider bell shapes and curves are also smoother. For prediction error results only small decreases are observed, however, smoother curves are also valid. These results are caused by the variable position setting being more resilient to occlusions since cameras are able to pick the object image majority of the time. This gives the networks enough intuition to generalize over action space.

In Figure 24, the Action Frame for variable object position setting and depth camera based location data results is presented. The results show high similarity with its counterpart (the fixed object position setting of the same experiment). An overall small improvement can be observed based on the added resilience mentioned above. Even so, we see a significant drop in accuracy for the sphere object at -90 degrees. Our analysis showed that this is again caused by the occlusions take place depending on the small size of the sphere object.

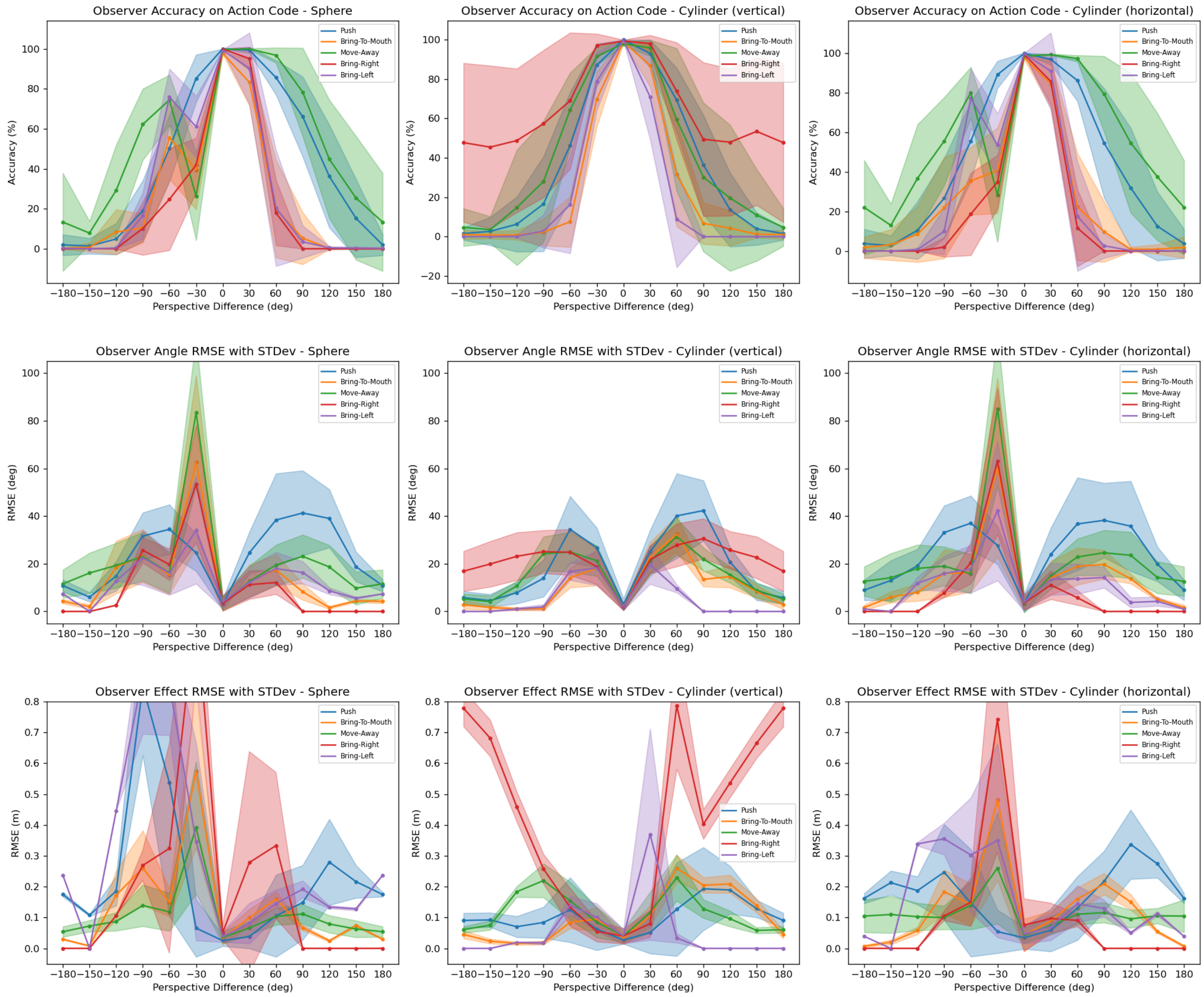

Figure 23: The Egocentric Variable under Fixed Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for depth camera based location data.

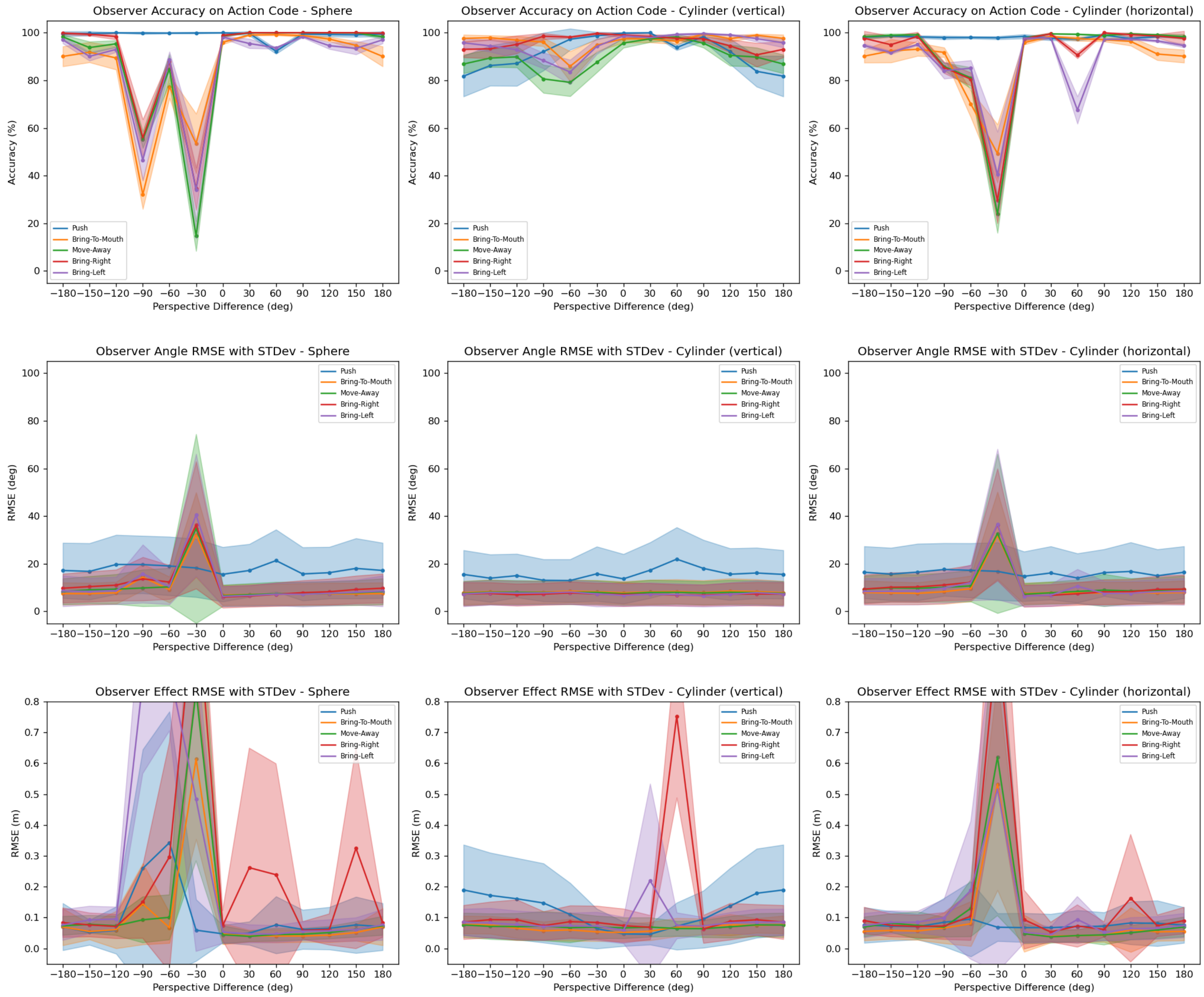

Figure 24: The Action Variable under Fixed Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for depth camera based location data.

6.2.3 Limitations of Using Depth Cameras

In this section, we discuss the limitations and issues of the current depth perception system that we employed in our experiments. Unlike the data provided by the simulator (center of the object), in its current form, the depth perception system returns the closest point's position on the object or the gripper that the camera sees as their position. This creates a bias for every observer differently since the closest point of the object or the gripper changes with perspective. Therefore, the precision of distance perception may change with position changes. Additionally, since every observer is represented with a Kinect camera in the simulation environment, using 12 Kinect cameras introduces computational complexity to the system. As mentioned in Chapter 4, images perceived by cameras sampled down with a factor of 8 to decrease the computational complexity. This may be a cause for more detection errors and decreased precision. However, further examination is not conducted about this subject. Moreover, occlusions that take place during the action cause the object and the gripper not able to be detected by the cameras (see Figure 25).

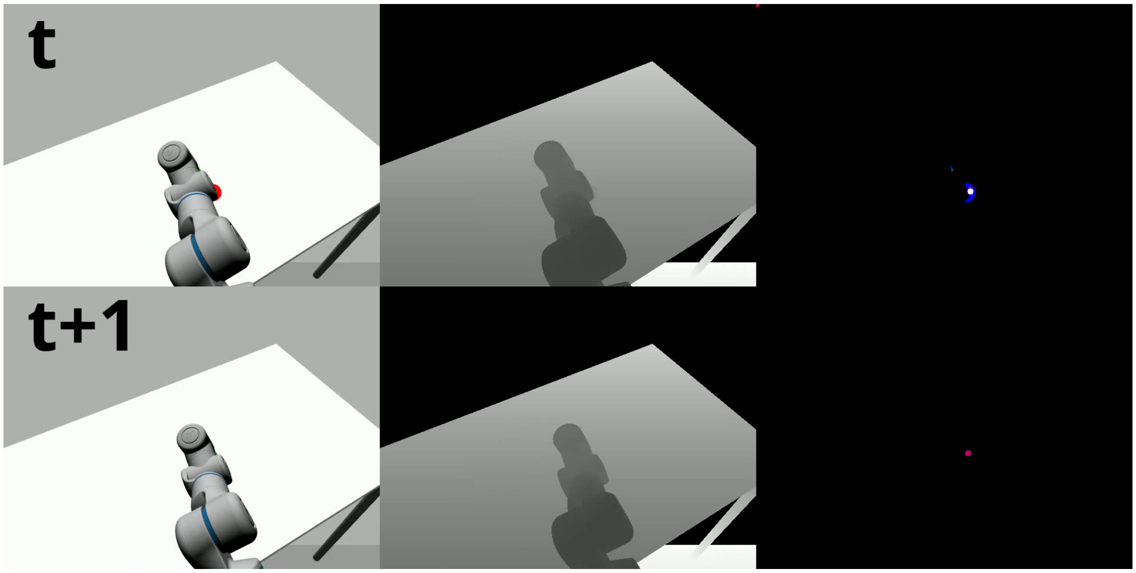

Figure 25: Illustrating two consecutive time steps where the gripper and the object are unable to be detected respectively because of the occlusions.

Furthermore, later in the development of the system (after the multi-object experiments), some issues with the simulation or ROS are caught. When action videos are examined it seems sudden jumps occurring. Even though the reason behind this issue is currently unknown, the effect it creates on the results is estimated as minimal. One reason for this is that, in its current form, the Action Frame only requires the object's first and last positions. However, the network requires five consecutive gripper positions to make predictions. When the target's position information is unable to be retrieved it is accepted as at the origin (0, 0, 0). Therefore, the action code prediction accuracy suffers from missing positional information as can be seen in Figure 22 under Subsection 6.2.1.

Table 1: Mean distance perception error of depth cameras with standard deviation in euclidean space(3D) for the robot gripper in the sphere experiments under the fixed setting.

Perspective Difference	**Push**	**Bring-To-Mouth**	**Move-Away**	**Bring-Left**	**Bring-Right**
-180°	1.03 ± 0.06	**7.23 ± 19.97**	0.81 ± 0.22	0.97 ± 0.11	0.95 ± 0.11
-150°	0.88 ± 0.1	0.79 ± 0.07	0.69 ± 0.15	0.79 ± 0.07	0.91 ± 0.15
-120°	0.76 ± 0.15	0.66 ± 0.06	0.61 ± 0.11	0.67 ± 0.09	0.85 ± 0.21
-90°	0.7 ± 0.17	0.56 ± 0.09	0.55 ± 0.1	0.61 ± 0.08	0.78 ± 0.21
-60°	0.7 ± 0.17	**2.27 ± 11.56**	**1.98 ± 10.46**	**4.78 ± 17.71**	**2.39 ± 11.25**
-30°	0.78 ± 0.17	**28.76 ± 37.49**	**33.17 ± 37.6**	**13.26 ± 28.67**	**19.98 ± 33.78**
0°	0.88 ± 0.12	**0.91 ± 5.63**	0.55 ± 0.1	0.63 ± 0.06	0.6 ± 0.07
30°	1.0 ± 0.06	0.57 ± 0.23	0.64 ± 0.15	0.74 ± 0.09	0.66 ± 0.14
60°	1.12 ± 0.07	0.7 ± 0.28	0.75 ± 0.23	0.91 ± 0.14	0.77 ± 0.22
90°	1.21 ± 0.13	0.84 ± 0.3	0.87 ± 0.28	1.08 ± 0.2	0.86 ± 0.29
120°	1.23 ± 0.15	0.96 ± 0.27	0.92 ± 0.31	1.16 ± 0.21	0.94 ± 0.29
150°	1.17 ± 0.11	**5.4 ± 16.65**	0.9 ± 0.28	1.12 ± 0.17	0.97 ± 0.21
180°	1.03 ± 0.06	**7.23 ± 19.97**	0.81 ± 0.22	0.97 ± 0.11	0.95 ± 0.11

The resulting position distance errors added to the system by these factors can be seen in Table 1 and 2 for the gripper and the sphere respectively under fixed object setting (For error tables related to the other object see Appendix C). The distance error is calculated by taking the mean and standard deviation of the difference of the simulator provided data and the data provided by the depth perception for all time steps of all samples. We choose 5 meters to replace the missing values (because of the occlusions) since we believe it could be a long distance considering all experiments occur on a table in a small area.

Table 2: Mean distance perception error of depth cameras with standard deviation in euclidean space(3D) for the sphere object under the fixed setting.

Perspective Difference	**Push**	**Bring-To-Mouth**	**Move-Away**	**Bring-Left**	**Bring-Right**
-180°	1.18 ± 0.03	1.15 ± 0.07	1.03 ± 0.18	1.21 ± 0.04	1.19 ± 0.04
-150°	1.16 ± 0.03	**1.18 ± 2.46**	1.0 ± 0.19	1.12 ± 0.05	1.23 ± 0.12
-120°	1.14 ± 1.16	**1.47 ± 6.12**	**1.27 ± 5.06**	**2.37 ± 10.37**	**1.76 ± 6.55**
-90°	**8.03 ± 22.42**	**9.18 ± 24.38**	**10.33 ± 25.79**	**31.58 ± 39.75**	**9.74 ± 24.69**
-60°	**8.56 ± 23.4**	**11.0 ± 26.76**	**18.57 ± 33.11**	**44.23 ± 40.47**	**14.44 ± 29.88**
-30°	0.91 ± 0.53	**44.68 ± 39.82**	**22.11 ± 34.68**	**24.99 ± 36.62**	**45.28 ± 40.61**
0°	0.9 ± 0.02	0.72 ± 0.21	0.8 ± 0.12	0.87 ± 0.04	**2.32 ± 10.8**
30°	0.91 ± 0.02	0.75 ± 1.08	0.81 ± 0.11	0.92 ± 0.02	**1.81 ± 8.87**
60°	0.96 ± 0.03	0.82 ± 1.11	0.85 ± 0.13	1.02 ± 0.09	**1.57 ± 7.29**
90°	1.01 ± 0.03	0.89 ± 0.5	0.91 ± 0.13	1.14 ± 0.17	0.94 ± 0.11
120°	1.08 ± 0.04	1.02 ± 0.67	0.97 ± 0.15	1.22 ± 0.18	1.02 ± 0.11
150°	1.14 ± 0.03	1.16 ± 1.85	1.01 ± 0.17	1.24 ± 0.13	1.26 ± 0.46
180°	1.18 ± 0.03	1.15 ± 0.07	1.03 ± 0.18	1.21 ± 0.04	1.19 ± 0.04

As can be seen in the Table 1 and 2, at certain angles the distance error increases. It can been seen that high distance error is common for -30 and -60 degrees in both tables. This explains the error spikes on the same offset on Figures 21 and 22. Essentially, for the **bring-to-mouth** action of sphere we see a drop in the accuracy at -180 and -150, when table 1 is observed, we see increased distance error for exact offset degrees.

CHAPTER VII

CONCLUSION AND FUTURE WORK

Mirror neurons found in the ventral premotor cortex of the primate brain encode goal directed actions in a multi-modal way, discharging both during the observation and execution of similar actions [5]. Most of these neurons show perspective invariant responses, thereby forming a basis for perspective invariant action understanding [66]. Although it is accepted that the brain employs several reference frames for interacting with the environment [3, 67], it is not clear whether the mirror neuron system is linked to a representation that uses a particular reference frame.

In this study, we proposed 'Action Frame' (AF) which is an ecologically and biologically plausible reference frame that represents action predictions and effects with respect to the gravity and the approach direction of a manipulator, i.e. hand. AF facilitates the development and learning of perspective invariant action recognition based on self-observation. Since primates are endowed with special neural circuits for visually processing hands [68] and representing gravity [3], AF may form the basis for mirror neuron system development.

To show the efficacy of AF in action understanding, firstly, we conducted experiments in a simulation environment with a humanoid robot equipped with the actions of **push, poke, bring-to-mouth** and **move-away** with a single-object (a vertically placed cylinder). Later, we expanded our experiments by adding two more objects (a sphere and a horizontally placed cylinder), also changed **poke** action with two different actions **bring-left** and **bring-right**. We conducted these experiments under two different settings as fixed and variable object positions to assess the AF performance also for different initial object positions and robot states.

In these experiments, we trained a prediction network based on the robot self-observation of executed actions by using either AF or EF based representations. Then, the network was asked to make predictions for observations from different perspectives. As expected, the prediction system based on EF gave a declining performance as the viewing angle deviates from the self-action view. In contrast, with AF, observed actions, their parameters, and the effect that would be generated could be accurately predicted. To verify that the results obtained were not simply due to the noise-free simulator data, similar experiments were conducted by using emulated depth-cameras to serve as the 'eyes' of the actor and the observers. Thus, the training and testing results were subject to the imperfections of the proof-of-concept visual processing employed. Although the results were affected by the imperfections, the capability of the perspective invariant action recognition system stayed at an acceptable level. This suggests that (1) The mirror neuron system may be the result of the development of a predictive system based on such a reference frame, and (2) the development of a predictive capability based on self-observation can be readily realized as part of a real robotic system.

From a neural network training perspective, the success of AF over EF is not really surprising as AF creates inputs that are similar (identical in the simulation environment). However it is critical to see the effect of perceptual noise and occlusions during real-world settings. Thus, our study not only computationally showed that AF type representation is powerful enough to sustain a mirror-like system but also showed the viability of its implementation on a robotic system.

Future work includes realizing the current experiment setup with a real robot and conducting the experiments presented in this thesis to show that robots can self-learn to recognize/understand human actions. To give the robot a general action recognition capability, the robot must be equipped with a richer action repertoire with dexterous manipulation capability. The visual acuity and thus recognition performance of the robot can be improved by employing state-of-the-art object detection and tracking methods such YOLO [69], instead of the color coded visual processing used in this thesis. Finally, a more fundamental approach to the problem considered in this thesis would be to investigate the emergence of Action Frame-like representations through end-to-end learning, rather than manually defining it. To this end, a straightforward approach would be to take a fixed action repertoire as in this thesis and implement the neural network as an encoder-decoder network where the bottleneck layer may be expected to form latent representations corresponding to objects and their spatial properties. Furthermore, a more elaborate approach can realize such an end-to-end learning in an action (reinforcement) learning setup where the output of the deep neural network represents action policy or action parameters.

APPENDIX A

THE ACTION FRAME RESULTS WITH SIMULATOR PROVIDED DATA

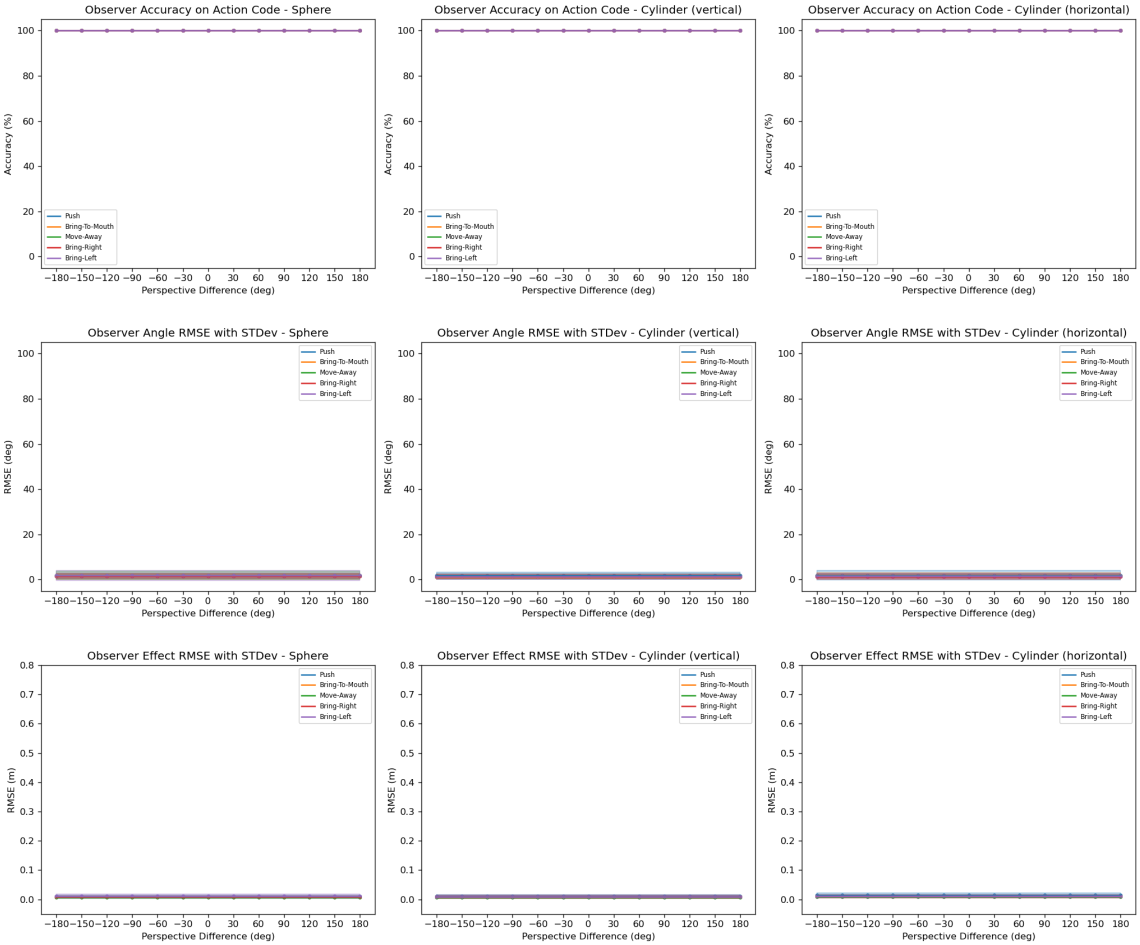

Figure 26: The Action Frame under Fixed Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for simulator provided location data.

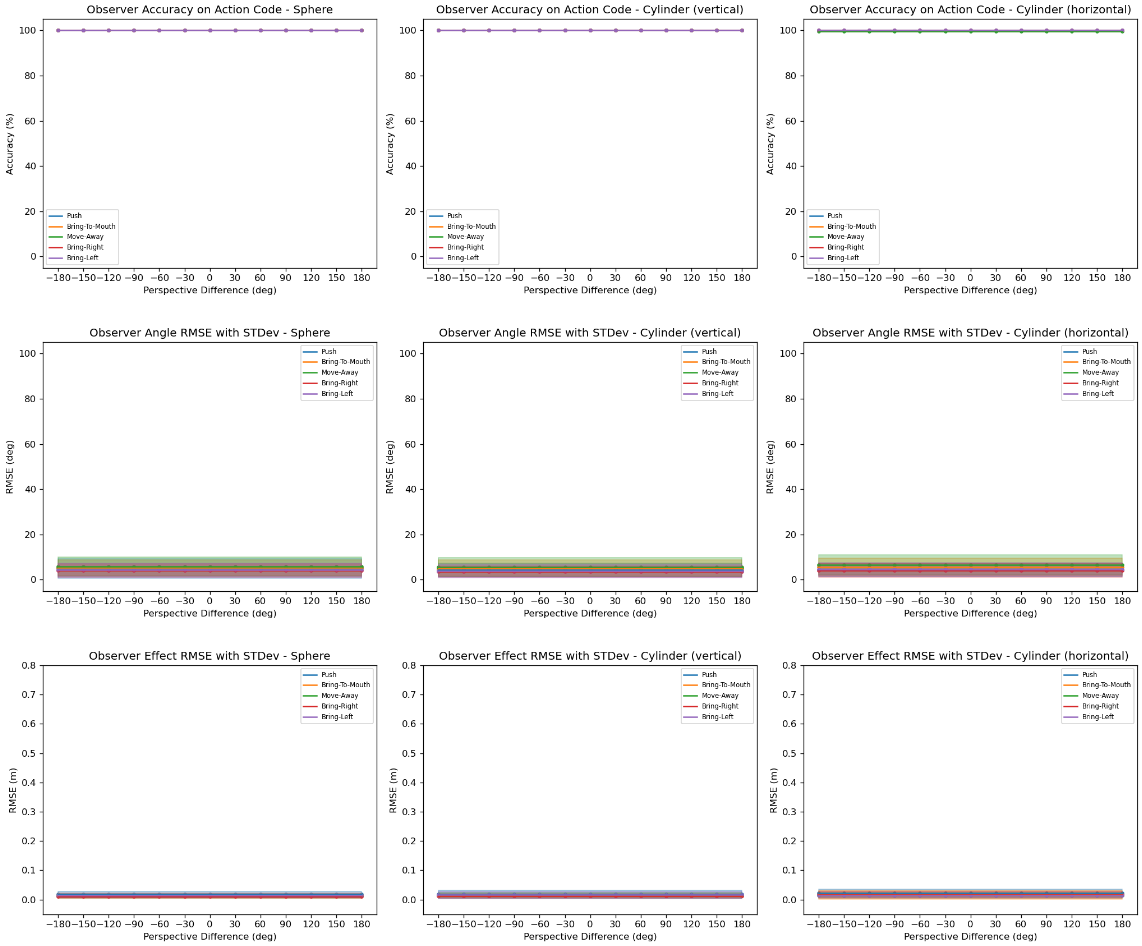

Figure 27: The Action Frame under Variable Object Position Setting showing action accuracy, angle prediction error and effect prediction error respectively as rows and for the objects as columns averaged over ten networks for simulator provided location data.

APPENDIX B

SAMPLE ACTION SEQUENCES

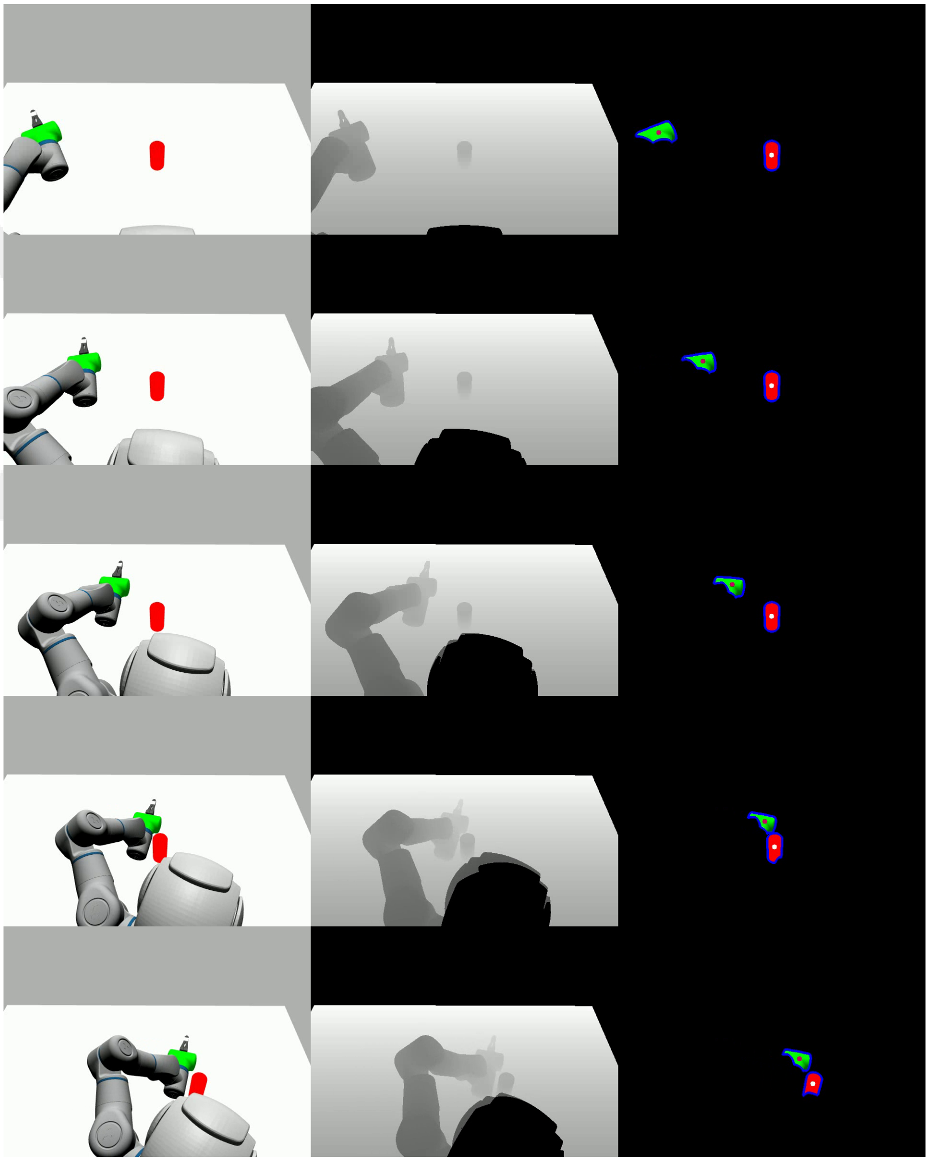

Figure 28: The **Push** action from actor's perspective at 0 degrees with vertical cylinder object under fixed position setting.

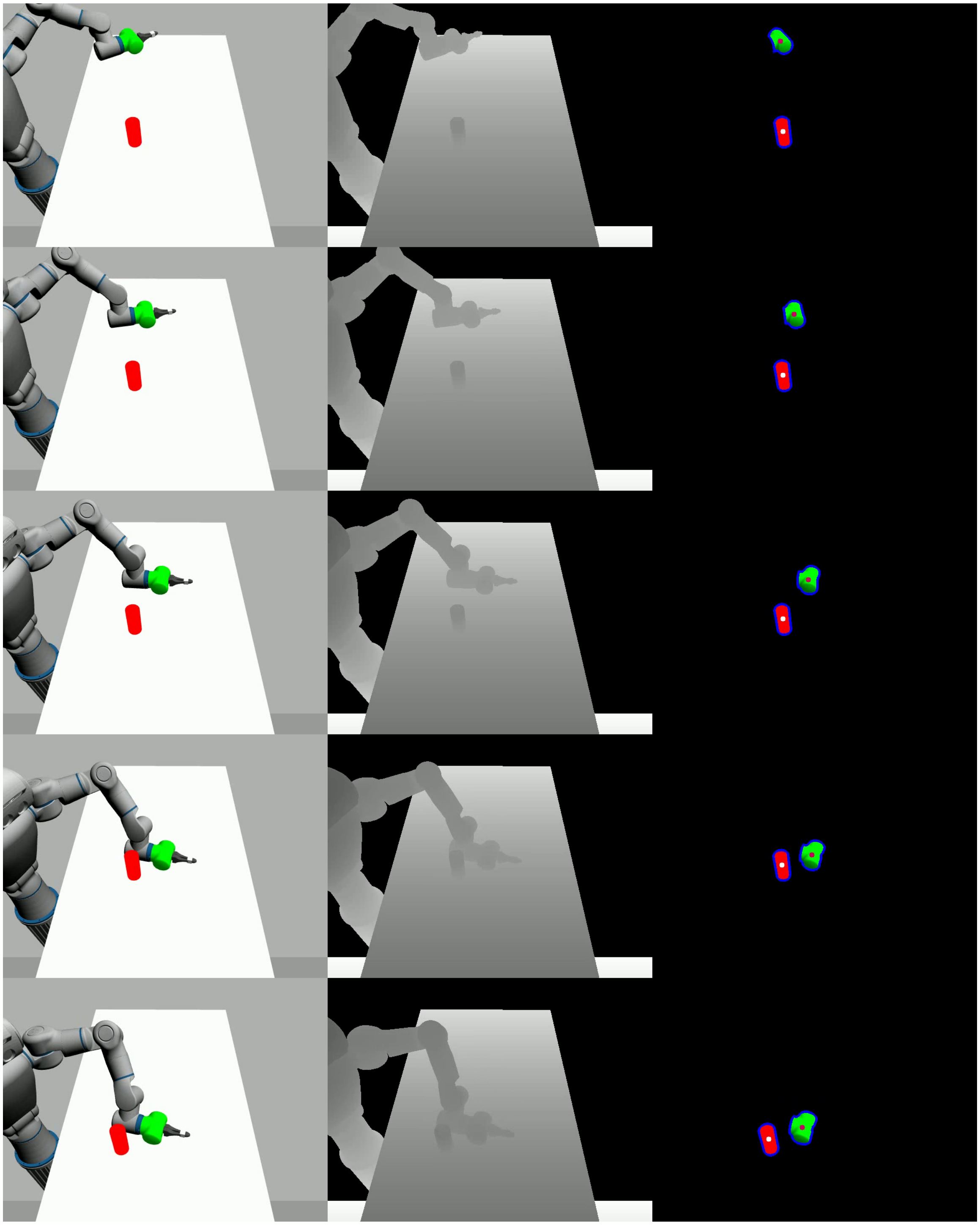

Figure 29: The **Push** action from observer's perspective at 90 degrees with vertical cylinder object under fixed position setting.

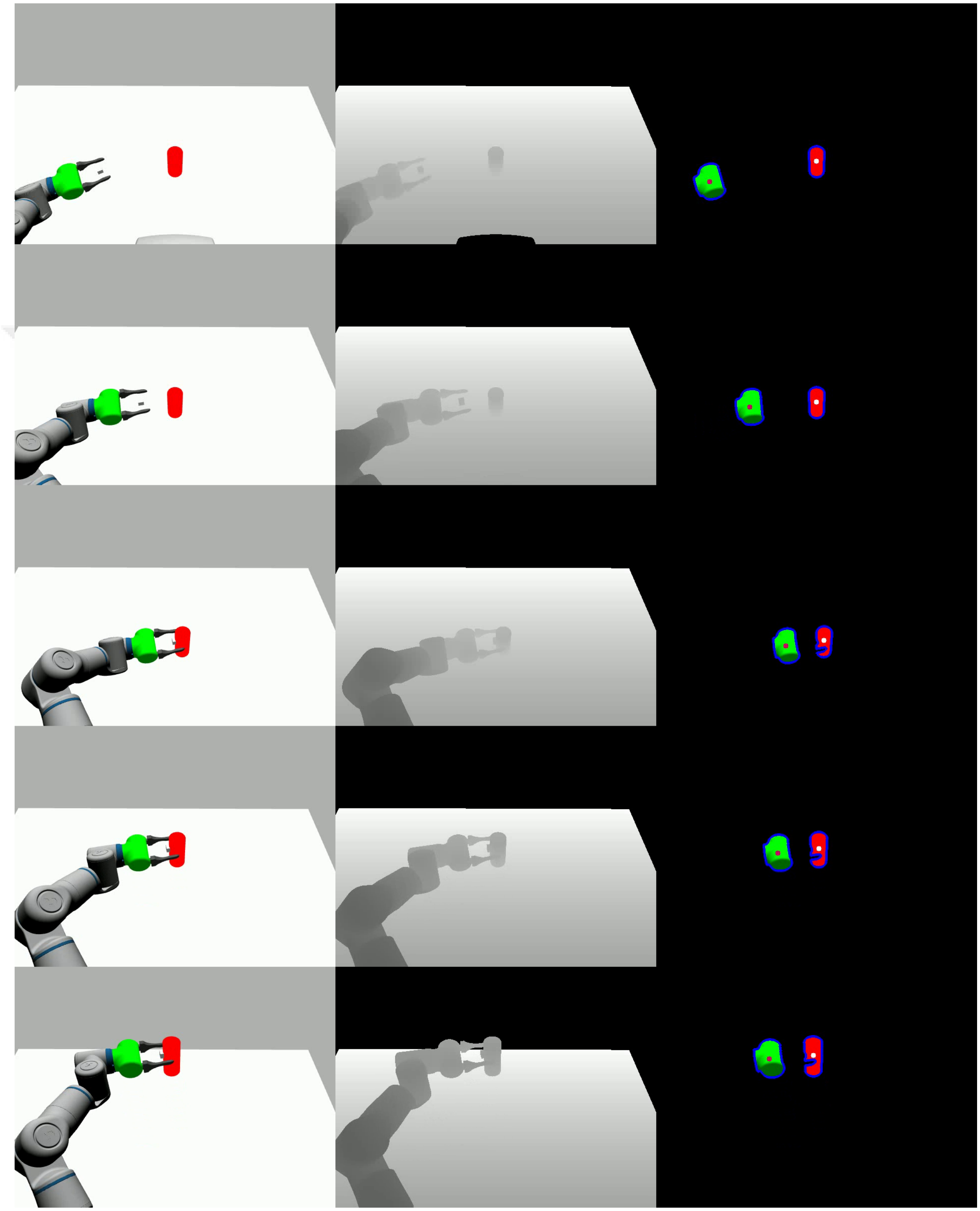

Figure 30: The **Move-Away** action from actor's perspective at 0 degrees with vertical cylinder object under fixed position setting.

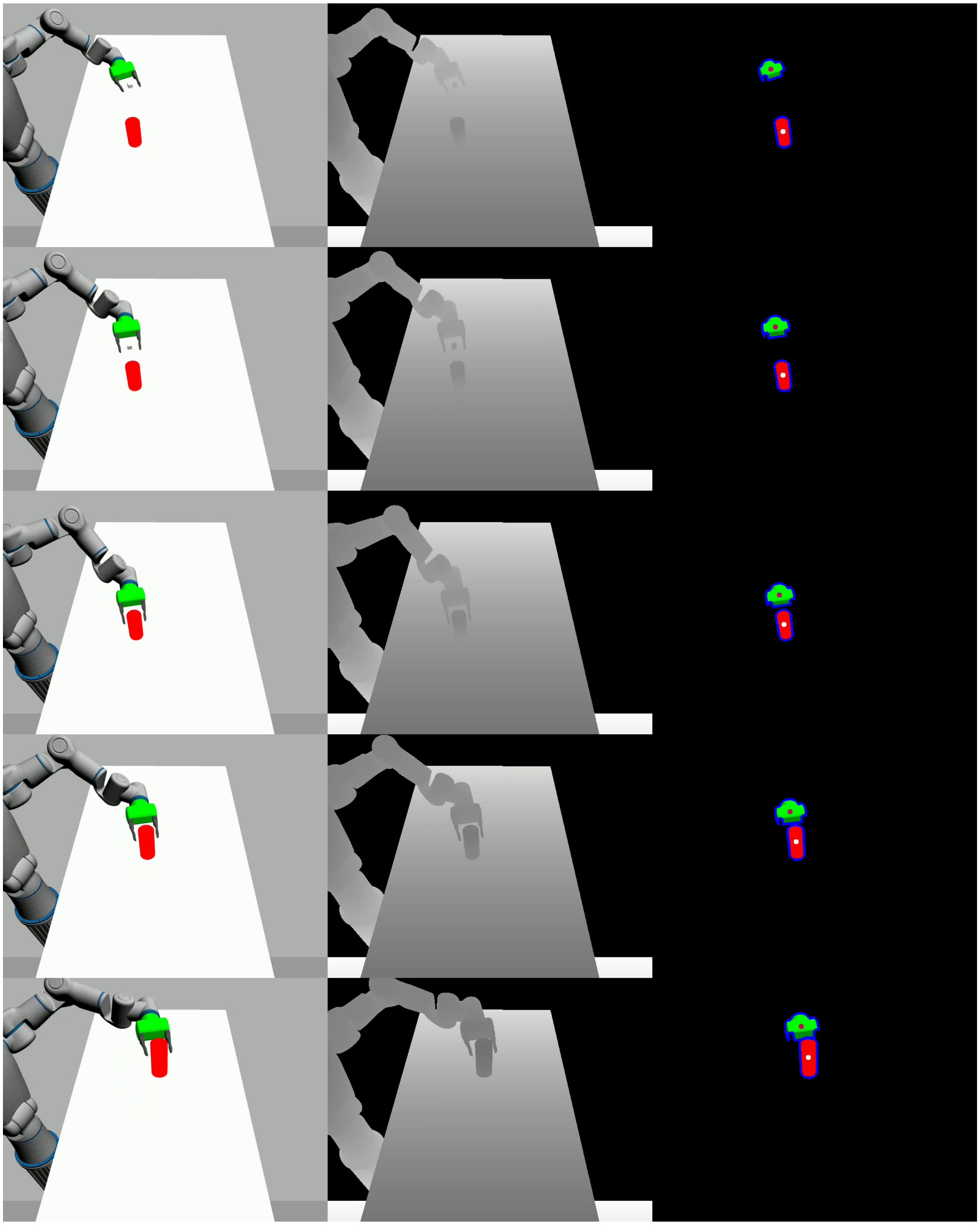

Figure 31: The **Move-Away** action from observer's perspective at 90 degrees with vertical cylinder object under fixed position setting.

APPENDIX C

DEPTH PERCEPTION ERROR TABLES

Table 3: Mean distance perception error of depth cameras with standard deviation in euclidean space(3D) for the robot gripper in the horizontal cylinder experiments under the fixed setting.

Perspective Difference	**Push**	**Bring-To-Mouth**	**Move-Away**	**Bring-Left**	**Bring-Right**
-180°	1.03 ± 0.06	**7.28 ± 20.05**	0.82 ± 0.22	0.95 ± 0.11	0.94 ± 0.11
-150°	0.88 ± 0.1	0.79 ± 0.07	0.7 ± 0.15	0.78 ± 0.08	0.89 ± 0.13
-120°	0.76 ± 0.15	0.66 ± 0.06	0.61 ± 0.11	0.66 ± 0.09	0.83 ± 0.2
-90°	0.7 ± 0.18	0.56 ± 0.09	0.56 ± 0.1	0.6 ± 0.09	0.76 ± 0.19
-60°	0.7 ± 0.18	**2.11 ± 11.04**	**2.39 ± 11.81**	**5.45 ± 19.01**	**2.22 ± 10.72**
-30°	0.79 ± 0.18	**28.27 ± 37.34**	**30.94 ± 37.22**	**13.69 ± 29.06**	**19.12 ± 33.24**
0°	0.88 ± 0.12	**0.96 ± 5.94**	0.56 ± 0.11	0.61 ± 0.06	0.59 ± 0.08
30°	1.0 ± 0.06	0.57 ± 0.23	0.66 ± 0.16	0.73 ± 0.09	0.65 ± 0.16
60°	1.2 ± 0.17	0.71 ± 0.28	0.77 ± 0.24	0.9 ± 0.14	0.76 ± 0.24
90°	1.21 ± 0.14	0.84 ± 0.3	0.89 ± 0.29	1.07 ± 0.2	0.85 ± 0.31
120°	1.23 ± 0.16	0.96 ± 0.27	0.95 ± 0.32	1.15 ± 0.21	0.93 ± 0.31
150°	1.16 ± 0.12	**5.4 ± 16.65**	0.92 ± 0.28	1.1 ± 0.17	0.96 ± 0.23
180°	1.03 ± 0.06	**7.28 ± 20.05**	0.82 ± 0.22	0.95 ± 0.11	0.94 ± 0.11

Table 4: Mean distance perception error of depth cameras with standard deviation in euclidean space(3D) for the horizontal cylinder experiments under the fixed setting.

Perspective Difference	**Push**	**Bring-To-Mouth**	**Move-Away**	**Bring-Left**	**Bring-Right**
-180°	1.18 ± 0.02	1.2 ± 1.99	1.07 ± 1.28	1.19 ± 0.07	1.19 ± 0.03
-150°	1.17 ± 0.04	**1.18 ± 2.25**	1.07 ± 1.75	1.11 ± 0.07	1.24 ± 0.13
-120°	1.13 ± 0.06	**1.12 ± 2.92**	1.05 ± 1.77	1.04 ± 0.12	1.23 ± 0.18
-90°	**1.32 ± 4.71**	**1.39 ± 6.37**	**1.19 ± 4.92**	**1.1 ± 3.6**	**1.21 ± 2.98**
-60°	**2.3 ± 10.17**	**3.15 ± 13.66**	**4.22 ± 16.15**	**4.42 ± 16.53**	**3.03 ± 12.64**
-30°	0.9 ± 0.03	**9.11 ± 24.43**	0.88 ± 1.87	0.86 ± 0.09	**12.73 ± 28.72**
0°	0.89 ± 0.03	**0.78 ± 2.21**	0.85 ± 1.78	0.86 ± 0.06	0.86 ± 0.06
30°	0.9 ± 0.02	**0.85 ± 3.03**	0.86 ± 1.76	0.91 ± 0.02	0.85 ± 0.09
60°	0.97 ± 0.05	**0.86 ± 2.14**	0.91 ± 1.89	0.99 ± 0.08	0.9 ± 0.11
90°	1.0 ± 0.04	0.88 ± 0.51	0.95 ± 1.72	1.11 ± 0.15	0.92 ± 0.13
120°	1.08 ± 0.04	1.02 ± 1.43	0.97 ± 0.16	1.2 ± 0.16	1.05 ± 0.26
150°	1.14 ± 0.02	**1.21 ± 2.74**	1.07 ± 1.71	1.23 ± 0.1	1.11 ± 0.13
180°	1.18 ± 0.02	1.2 ± 1.99	1.07 ± 1.28	1.19 ± 0.07	1.19 ± 0.03

Table 5: Mean distance perception error of depth cameras with standard deviation in euclidean space(3D) for the robot gripper in the vertical cylinder experiments under the fixed setting.

Perspective Difference	**Push**	**Bring-To-Mouth**	**Move-Away**	**Bring-Left**	**Bring-Right**
-180°	1.02 ± 0.06	1.1 ± 0.08	0.98 ± 0.18	1.13 ± 0.06	1.12 ± 0.06
-150°	0.88 ± 0.09	0.93 ± 0.04	0.84 ± 0.12	0.93 ± 0.04	1.05 ± 0.15
-120°	0.76 ± 0.15	0.78 ± 0.04	0.74 ± 0.09	0.79 ± 0.07	0.97 ± 0.22
-90°	0.7 ± 0.17	0.68 ± 0.08	0.68 ± 0.08	**8.42 ± 24.37**	0.89 ± 0.21
-60°	0.7 ± 0.17	0.63 ± 0.11	0.66 ± 0.07	**12.02 ± 28.3**	0.84 ± 0.19
-30°	0.79 ± 0.17	0.6 ± 0.13	0.65 ± 0.06	0.7 ± 0.04	0.75 ± 0.08
0°	0.88 ± 0.12	0.63 ± 0.16	0.7 ± 0.07	0.75 ± 0.03	0.75 ± 0.03
30°	0.99 ± 0.05	0.73 ± 0.21	0.81 ± 0.11	0.9 ± 0.05	0.81 ± 0.11
60°	1.12 ± 0.07	0.89 ± 0.25	0.95 ± 0.18	1.11 ± 0.11	0.95 ± 0.18
90°	1.21 ± 0.13	1.06 ± 0.24	1.09 ± 0.23	1.31 ± 0.18	1.04 ± 0.26
120°	1.23 ± 0.15	1.19 ± 0.2	1.15 ± 0.25	1.4 ± 0.19	1.13 ± 0.25
150°	1.16 ± 0.11	1.2 ± 0.14	1.11 ± 0.24	1.32 ± 0.14	1.16 ± 0.16
180°	1.02 ± 0.06	1.1 ± 0.08	0.98 ± 0.18	1.13 ± 0.06	1.12 ± 0.06

Table 6: Mean distance perception error of depth cameras with standard deviation in euclidean space(3D) for the vertical cylinder experiments under the fixed setting.

Perspective Difference	**Push**	**Bring-To-Mouth**	**Move-Away**	**Bring-Left**	**Bring-Right**
-180°	1.1 ± 0.03	1.11 ± 0.06	0.99 ± 0.14	1.15 ± 0.06	1.15 ± 0.05
-150°	1.09 ± 0.03	1.07 ± 0.04	0.97 ± 0.14	1.06 ± 0.04	1.2 ± 0.15
-120°	1.04 ± 0.05	0.98 ± 0.05	0.94 ± 0.12	0.97 ± 0.09	1.21 ± 0.23
-90°	0.98 ± 0.06	0.87 ± 0.1	0.87 ± 0.1	0.88 ± 0.1	1.12 ± 0.22
-60°	0.91 ± 0.06	0.76 ± 0.14	0.81 ± 0.08	0.83 ± 0.08	0.99 ± 0.15
-30°	0.85 ± 0.04	0.71 ± 0.16	0.77 ± 0.08	0.8 ± 0.06	0.88 ± 0.06
0°	0.83 ± 0.02	0.69 ± 0.17	0.75 ± 0.08	0.81 ± 0.03	0.81 ± 0.02
30°	0.85 ± 0.02	0.71 ± 0.18	0.77 ± 0.09	**1.8 ± 8.18**	0.8 ± 0.06
60°	0.88 ± 0.03	0.77 ± 0.15	0.81 ± 0.1	0.96 ± 0.12	**3.91 ± 15.54**
90°	0.95 ± 0.04	0.86 ± 0.12	0.87 ± 0.11	1.09 ± 0.18	0.88 ± 0.1
120°	1.02 ± 0.05	0.98 ± 0.49	0.94 ± 0.13	1.18 ± 0.19	0.96 ± 0.1
150°	1.08 ± 0.04	1.08 ± 0.04	0.98 ± 0.14	1.2 ± 0.14	1.06 ± 0.05
180°	1.1 ± 0.03	1.11 ± 0.06	0.99 ± 0.14	1.15 ± 0.06	1.15 ± 0.05

Bibliography

[1] R. Peeters, L. Simone, K. Nelissen, M. Fabbri-Destro, W. Vanduffel, G. Rizzolatti, and G. A. Orban, "The representation of tool use in humans and monkeys: Common and uniquely human features," *Journal of Neuroscience*, vol. 29, no. 37, pp. 11523–11539, 2009.

[2] D. I. Perrett, M. H. Harries, R. Bevan, S. Thomas, P. J. Benson, A. J. Mistlin, A. J. Chitty, J. K. Hietanen, and J. E. Ortega, "Frameworks of analysis for the neural representation of animate objects and actions," *Journal of Experimental Biology*, vol. 146, no. 1, pp. 87–113, 1989.

[3] A. Rosenberg and D. E. Angelaki, "Gravity influences the visual representation of object tilt in parietal cortex," *Journal of Neuroscience*, vol. 34, no. 43, pp. 14170–14180, 2014.

[4] M. Oram and D. Perrett, "Responses of anterior superior temporal polysensory (stpa) neurons to "biological motion" stimuli," *Journal of cognitive neuroscience*, vol. 6, no. 2, pp. 99–116, 1994.

[5] G. Di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti, "Understanding motor events: a neurophysiological study," *Experimental brain research*, vol. 91, no. 1, pp. 176–180, 1992.

[6] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi, "Premotor cortex and the recognition of motor actions," *Cognitive Brain Research*, vol. 3, no. 2, pp. 131–141, 1996. Mental representations of motor acts.

[7] A. Compston, "Action recognition in the premotor cortex. By Vittorio Gallese, Luciano Fadiga, Leonardo Fogassi and Giacomo Rizzolatti. Brain 1996: 119; 593–609.," *Brain*, vol. 132, pp. 1685–1689, 06 2009.

[8] L. Fogassi, P. F. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, and G. Rizzolatti, "Parietal lobe: from action organization to intention understanding," *Science*, vol. 308, no. 5722, pp. 662–667, 2005.

[9] S. Rozzi, P. F. Ferrari, L. Bonini, G. Rizzolatti, and L. Fogassi, "Functional organization of inferior parietal lobule convexity in the macaque monkey: electrophysiological characterization of motor, sensory and mirror responses and their correlation with cytoarchitectonic areas," *European Journal of Neuroscience*, vol. 28, no. 8, pp. 1569–1588, 2008.

[10] G. Rizzolatti and L. Fadiga, "Grasping objects and grasping action meanings: the dual role of monkey rostroventral premotor cortex (area f5)," in *Novartis Foundation Symposium*, pp. 81–94, Wiley Online Library, 1998.

[11] A. Murata, L. Fadiga, L. Fogassi, V. Gallese, V. Raos, and G. Rizzolatti, "Object representation in the ventral premotor cortex (area f5) of the monkey," *Journal of neurophysiology*, vol. 78, no. 4, pp. 2226–2230, 1997.

[12] E. Kohler, C. Keysers, M. A. Umilta, L. Fogassi, V. Gallese, and G. Rizzolatti, "Hearing sounds, understanding actions: action representation in mirror neurons," *Science*, vol. 297, no. 5582, pp. 846–848, 2002.

[13] R. Mukamel, A. D. Ekstrom, J. Kaplan, M. Iacoboni, and I. Fried, "Single-neuron responses in humans during execution and observation of actions," *Current biology*, vol. 20, no. 8, pp. 750–756, 2010.

[14] P. Molenberghs, R. Cunnington, and J. B. Mattingley, "Brain regions with mirror properties: a meta-analysis of 125 human fmri studies," *Neuroscience & Biobehavioral Reviews*, vol. 36, no. 1, pp. 341–349, 2012.

[15] G. Rizzolatti and L. Craighero, "The mirror-neuron system," *Annu. Rev. Neurosci.*, vol. 27, pp. 169–192, 2004.

[16] M. Umiltà, E. Kohler, V. Gallese, L. Fogassi, L. Fadiga, C. Keysers, and G. Rizzolatti, "I know what you are doing: A neurophysiological study," *Neuron*, vol. 31, no. 1, pp. 155–165, 2001.

[17] G. Rizzolatti, L. Cattaneo, M. Fabbri-Destro, and S. Rozzi, "Cortical mechanisms underlying the organization of goal-directed actions and mirror neuron-based action understanding," *Physiological Reviews*, vol. 94, no. 2, pp. 655–706, 2014. PMID: 24692357.

[18] G. Rizzolatti, L. Fogassi, and V. Gallese, "Neurophysiological mechanisms underlying the understanding and imitation of action," *Nature reviews neuroscience*, vol. 2, no. 9, pp. 661–670, 2001.

[19] M. A. Arbib and J. Bonaiuto, "From grasping to complex imitation: mirror systems on the path to language," *Mind & Society*, vol. 7, no. 1, pp. 43–64, 2008.

[20] M. C. Corballis, "Mirror neurons and the evolution of language," *Brain and Language*, vol. 112, no. 1, pp. 25–35, 2010. Mirror Neurons: Prospects and Problems for the Neurobiology of Language.

[21] G. Rizzolatti and M. A. Arbib, "Language within our grasp," *Trends in Neurosciences*, vol. 21, no. 5, pp. 188–194, 1998.

[22] K. Nelissen, E. Borra, M. Gerbella, S. Rozzi, G. Luppino, W. Vanduffel, G. Rizzolatti, and G. A. Orban, "Action observation circuits in the macaque monkey cortex," *Journal of Neuroscience*, vol. 31, no. 10, pp. 3743–3756, 2011.

[23] D. Perrett, M. Oram, M. Harries, R. Bevan, J. Hietanen, P. Benson, and S. Thomas, "Viewer-centred and object-centred coding of heads in the macaque temporal cortex," *Experimental brain research*, vol. 86, no. 1, pp. 159–173, 1991.

[24] V. Caggiano, L. Fogassi, G. Rizzolatti, J. K. Pomper, P. Thier, M. A. Giese, and A. Casile, "View-based encoding of actions in mirror neurons of area f5 in macaque premotor cortex," *Current Biology*, vol. 21, no. 2, pp. 144–148, 2011.

[25] D. I. Perrett, A. J. Mistlin, and A. J. Chitty, "Visual neurones responsive to faces," *Trends in Neurosciences*, vol. 10, no. 9, pp. 358–364, 1987.

[26] S. Rozzi, R. Calzavara, A. Belmalih, E. Borra, G. G. Gregoriou, M. Matelli, and G. Luppino, "Cortical connections of the inferior parietal cortical convexity of the macaque monkey," *Cerebral Cortex*, vol. 16, no. 10, pp. 1389–1417, 2006.

[27] G. Rizzolatti and L. Fogassi, "The mirror mechanism: recent findings and perspectives," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1644, p. 20130420, 2014.

[28] C. L. Colby, J.-R. Duhamel, and M. E. Goldberg, "Visual, presaccadic, and cognitive activation of single neurons in monkey lateral intraparietal area," *Journal of neurophysiology*, vol. 76, no. 5, pp. 2841–2852, 1996.

[29] C. L. Colby and J.-R. Duhamel, "Heterogeneity of extrastriate visual areas and multiple parietal areas in the macaque monkey," *Neuropsychologia*, vol. 29, no. 6, pp. 517–537, 1991.

[30] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[31] D. O. Hebb, *The organization of behavior: A neuropsychological theory.* Psychology Press, 2005.

[32] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[34] D. J. Finney, *Probit analysis: a statistical treatment of the sigmoid response curve.* Cambridge university press, Cambridge, 1952.

[35] E. Fan, "Extended tanh-function method and its applications to nonlinear equations," *Physics Letters A*, vol. 277, no. 4, pp. 212–218, 2000.

[36] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines.," in *ICML* (J. Fürnkranz and T. Joachims, eds.), pp. 807–814, Omnipress, 2010.

[37] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[39] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[40] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and vision computing*, vol. 60, pp. 4–21, 2017.

[41] N. Jaouedi, N. Boujnah, O. Htiwich, and M. S. Bouhlel, "Human action recognition to human behavior analysis," in *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pp. 263–266, IEEE, 2016.

[42] Y. Ji, Y. Yang, F. Shen, H. T. Shen, and X. Li, "A survey of human action analysis in hri applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2114–2128, 2019.

[43] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3d hand pose estimation from monocular video," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1793–1805, 2011.

[44] C. Choi, S. Ho Yoon, C.-N. Chen, and K. Ramani, "Robust hand pose estimation during the interaction with an unknown object," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3123–3132, 2017.

[45] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *European Conference on Computer Vision*, pp. 852–863, Springer, 2012.

[46] S. Singh, C. Arora, and C. V. Jawahar, "First person action recognition using deep learned descriptors," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2620–2628, 2016.

[47] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*, pp. 319–345. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[48] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1894–1903, 2016.

[49] G. Garcia-Hernando, S. Yuan, S. Baek, and T. Kim, "First-person hand action benchmark with RGB-D videos and 3d hand pose annotations," *CoRR*, vol. abs/1704.02463, 2017.

[50] Y. Takahashi, Y. Tamura, M. Asada, and M. Negrello, "Emulation and behavior understanding through shared values," *Robotics and Autonomous Systems*, vol. 58, no. 7, pp. 855–865, 2010. Advances in Autonomous Robots for Service and Entertainment.

[51] S. Sempena, Nur Ulfa Maulidevi, and Peb Ruswono Aryan, "Human action recognition using dynamic time warping," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pp. 1–5, 2011.

[52] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

[53] L. Bonini, M. Maranesi, A. Livi, L. Fogassi, and G. Rizzolatti, "Space-dependent representation of objects and other's action in monkey ventral premotor grasping neurons," *Journal of Neuroscience*, vol. 34, no. 11, pp. 4108–4119, 2014.

[54] E. Oztop, M. Kawato, and M. Arbib, "Mirror neurons and imitation: a computationally guided review," *Neural Networks*, vol. 19, no. 3, pp. 254–71, 2006.

[55] E. Oztop, M. Kawato, and M. A. Arbib, "Mirror neurons: Functions, mechanisms and models," *Neuroscience Letters*, vol. 540, pp. 43–55, 2013.

[56] E. Oztop and M. A. Arbib, "Schema design and implementation of the grasp-related mirror neuron system," *Biological cybernetics*, vol. 87, no. 2, pp. 116–140, 2002.

[57] T. Chaminade, E. Oztop, G. Cheng, and M. Kawato, "From self-observation to imitation: Visuomotor association on a robotic hand," *Brain research bulletin*, vol. 75, no. 6, pp. 775–784, 2008.

[58] F. Dawood and C. K. Loo, "View-invariant visuomotor processing in computational mirror neuron system for humanoid," *PloS one*, vol. 11, no. 3, p. e0152003, 2016.

[59] J. Bonaiuto and M. A. Arbib, "Extending the mirror neuron system model, ii: what did i just do? a new role for mirror neurons," *Biological cybernetics*, vol. 102, no. 4, pp. 341–359, 2010.

[60] Y. Nagai, Y. Kawai, and M. Asada, "Emergence of mirror neuron system: Immature vision leads to self-other correspondence," in *2011 IEEE International Conference on Development and Learning (ICDL)*, vol. 2, pp. 1–6, IEEE, 2011.

[61] Y. Demiris* and M. Johnson†, "Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning," *Connection Science*, vol. 15, no. 4, pp. 231–243, 2003.

[62] G. Metta, G. Sandini, L. Natale, L. Craighero, and L. Fadiga, "Understanding mirror neurons: A bio-robotic approach," *Interaction Studies*, vol. 7, no. 2, pp. 197–232, 2006.

[63] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149–2154 vol.3, 2004.

[64] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system."

[65] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[66] G. Rizzolatti, L. Fogassi, and V. Gallese, "Neurophysiological mechanisms underlying the understanding and imitation of action," *Nature Reviews Neuroscience*, vol. 2, no. 9, pp. 661–70, 2001.

[67] C. L. Colby, "Action-oriented spatial reference frames in cortex," *Neuron*, vol. 20, no. 1, pp. 15–24, 1998.

[68] T. Jellema, C. Baker, B. Wicker, and D. Perrett, "Neural representation for the perception of the intentionality of actions," *Brain and Cognition*, vol. 44, no. 2, pp. 280–302, 2000.

[69] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.