Active Observers in a 3D World: Human Visual Behaviours for Active Vision

Markus D. Solbach

A Dissertation Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Graduate Program in
Electrical Engineering and Computer Science
York University
Toronto, Ontario

July 2022

Abstract

Human-like performance in computational vision systems is yet to be achieved. In fact, human-like visuospatial behaviours are not well understood – a crucial capability for any robotic system whose role is to be a real assistant. This dissertation examines human visual behaviours involved in solving a well-known visual task; *The Same-Different Task*. It is used as a probe to explore the space of active human observation during visual problem-solving. It asks a simple question: "are two objects the same?". To study this question, we created a set of novel objects with known complexity to push the boundaries of the human visual system. We wanted to examine these behaviours as opposed to the static, 2D, display-driven experiments done to date. We thus needed to develop a complete infrastructure for an experimental investigation using 3D objects and active, free, human observers. We have built a novel, psychophysical experimental setup that allows for precise and synchronized gaze and head-pose tracking to analyze subjects performing the task. To the best of our knowledge, no other system provides the same characteristics. We have collected detailed, first-of-its-kind data of humans performing a visuospatial task in hundreds of experiments. We present an in-depth analysis of different metrics of humans solving this task, who demonstrated up to 100% accuracy for specific settings and that no trial used less than six fixations. We provide a complexity analysis that reveals human performance in solving this task is about $\mathcal{O}(n)$, where n is the size of the object. Furthermore, we discovered that our subjects used many different visuospatial strategies and showed that they are deployed dynamically. Strikingly, no learning effect was observed that affected the accuracy. With this extensive and unique data set, we addressed its computational counterpart. We used reinforcement learning to learn the three-dimensional same-different task and discovered crucial limitations which only were overcome if the task was simplified to the point of trivialization. Lastly, we formalized a set of suggestions to inform the enhancement of existing

machine learning methods based on our findings from the human experiments and multiple tests we performed with modern machine learning methods.

To my mother and father,

Maria Anna and Dieter,

for their wholehearted love and support.

Für Mutter und Vater,

Maria Anna und Dieter,

für ihre vom ganzen Herzen kommende Liebe.

Acknowledgments

The work presented here only exists because of the support of many.

First and foremost, I would like to thank my supervisor, John K. Tsotsos, who is an outstanding scientific guide to me. A person I can only describe with the highest attributes. John, you made my Ph.D. studies at your lab immensely enjoyable. Your knowledge, guidance, kindness and the role model you are, not only as a scientist, inspired and will inspire me for the rest of my life.

Thank you to all of my labmates and collaborators with whom I had the pleasure to spend time and work. All of you have a significant stake in this work, especially by making the Tsotsos lab a place to grow.

My family has given me the support needed to be where I am today. To my wife, Gabriela, who is an invaluable pillar in my life. Your love, support and advice kept me steady on my journey to achieve this goal. I love you. To my parents, who taught me to be curious and were always amused when I opened up yet another device to "check" how it worked on the inside. (Typically, the device was broken afterwards.) To my sister, Anke, who fostered my interest in mathematics and helped lay the groundwork for this degree. To Dori, who is a better person than I am even though she is a dog. Your constant warmth and companionship I could never repay. But I will try anyway.

To my examination committee, Richard P. Wildes, Michael S. Brown, Jeffrey D. Schall and Henrik I. Christensen, for your time, ideas and valuable feedback. You have helped improve my work and this dissertation in many ways. I am deeply grateful that you were part of this.

Finally, this is not a conclusive acknowledgement, as many others have affected this document. I hope all of you know how priceless your input and the conversations we had were to me.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
AP	Active-Pentuple
API	Application programming interface
CAD	Computer-aided design
CP	Cognitive Program
CV	Computer Vision
CMU	Carnegie Mellon University
CNN	Convolutional neural networks
CSV	Comma-separated values
CUHK	The Chinese University of Hong Kong
DNN	Deep Neural Network
DOF	Degrees of Freedom
EEG	Electroencephalography
HCI	Human-Computer Interaction
HOG	Histogram of oriented gradients
HRI	Human-Robot Interaction
ICCV	International Conference on Computer Vision
IMU	Inertial measurement unit
lib	Library
LIDAR	Light Detection and Ranging
MDP	Markov decision processes
MIT	Massachusetts Institute of Technology

ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
MS-COCO	Microsoft Common Objects in Context
Net	Network
$\mathcal{O}$	Big O complexity class
PASCAL VOC	The PASCAL Visual Object Classes
PESAO	Psychophysical Experimental Setup for Active Observers
POMDP	Partially observable MDP
PPO	Proximal Policy Optimization
PTS	Presentation Timestamp
R-CNN	Region Based Convolutional Neural Networks
ResNet	Residual Network
RGB	Red Green Blue colour space
RGB-D	Red Green Blue colour space plus depth
RL	Reinforcement Learning
SAC	Soft Actor-Critic
SD	Same-Different
SDK	Software development kit
SIFT	Scale-invariant feature transform
SLAM	Simultaneous localization and mapping
STL	Stereolithography
SVM	Support vector machines
SVRT	Synthetic Visual Reasoning Test
TEOS	The Effect of Self-Occlusion
VEDB	Visual experience database
VGG	Visual Geometry Group model
VR	Visual Routines
XDF	Extensible Data Format
2D	Two-Dimensional

3D	Three-Dimensional
6D	Six-Dimensional

Chapter 1

Introduction

1.1 Background

Vision is described as "the *process* of discovering from images what is present in the world, and where it is" [Marr, 1982] or as put by Aristotle, "to know what is where by looking" [Barnes, 1995].

Talos, a bronze giant of classical mythology, was made by the ancient god Hephaestus, the god of fire and iron, on order of Zeus and was given as a gift to King Minos of the island of Crete [Graves, 1993]. Talos is the earliest mention of an artificial visually-guided agent. It can be viewed as an ancient Greek humanoid robot whose purpose was to protect the island from pirates and invaders, ensuring that the laws of the island were upheld by circling the island three times a day. Figure 1.1 shows Talos as depicted in the film "Jason and the Argonauts" (1963). This is just one example of how artificial vision systems have fascinated humans throughout history.

Figure 1.1: Talos, a bronze giant of classical mythology. Here, Talos is portraited in the film "Jason and the Argonauts" (1963). It is the earliest mention of an artificial visually-guided agent. Source: Still from Chaffey (1963), Jason and the Argonauts, 0:35:35, Charles H. Schneer Productions

Computer vision is the scientific discipline that encompasses the efforts to create systems that gain a high-level understanding from digital images. The scientific community uses inspirations from different sources, for example the primate visual system [Tsotsos, 1987, Itti et al., 1998, Alleysson

et al., 2005], other biological visual systems [Ballard, 1991, Terzopoulos and Rabie, 1995, Jhuang, 2007, LeCun et al., 2010], and first principles from physics and mathematics [Horn et al., 1986].

Nevertheless, where did modern computer vision research start? One of the earliest applications was to recognize patterns in order to identify characters in documents for office automation-related tasks [Roberts, 1960, Tippett et al., 1965]. [Roberts, 1963] first used what is now a common technique in computer vision, the matching of two-dimensional image features with three-dimensional representations of objects. Figure 1.2 illustrates some of the steps of this procedure.

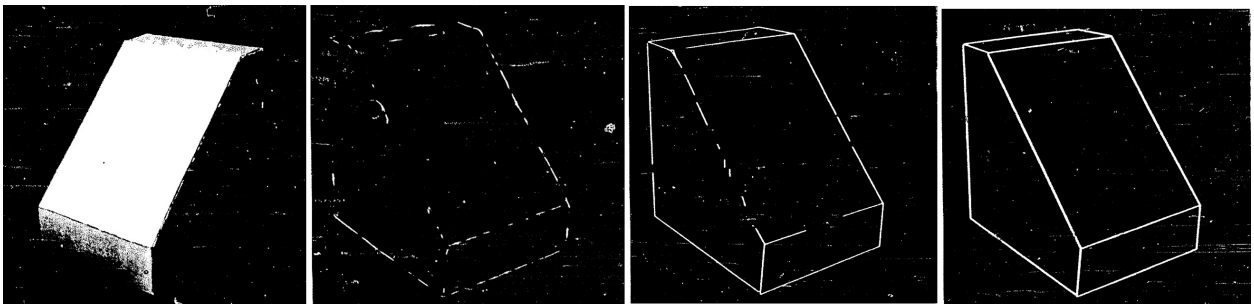


Figure 1.2: The first use of two-dimensional image feature matching with three-dimensional object representations. Left to right: Original Picture, Connected Feature Points, Initial Like Fitting, Final Line Drawing. Adapted from: [Roberts, 1963]

Gerald Jay Sussman, a first-year undergraduate, was given the task by Marvin Minsky at MIT in 1966 to "spend the summer linking a camera to a computer and getting the computer to describe what it saw." [McKevitt, 1997, Boden, 2013]. This was not a joke, as both Minsky and Sussman expected the project to succeed. As it turned out, it was not a summer-long project and decades later, the interest in solving the problem to "know what is where by looking" with a computer, is unabated.

The computer vision community, working on sub-domains such as object recognition, object localization, face detection, pose estimation, and scene understanding – to name a few – demonstrated remarkable progress in solving various aspects of this problem [Lowe, 1999, Viola and Jones, 2004, Michel et al., 2017, Li et al., 2009, Lampert et al., 2008].

Image data is the most common input modality for computer vision. Others are, for example, RGB-D images and LIDAR data. Tremendous progress was possible with the availability of large-scale data sets. *ImageNet* [Deng et al., 2009] is an image data set collected from the web with hundreds and thousands of examples per node of the corresponding *WordNet* hierarchy [Fellbaum,

2010]. The data sets consist of thousands of classes and over 14 million images and is constantly growing. Other examples of large scale image data sets are Microsoft COCO [Lin et al., 2014], PASCAL VOC [Everingham et al., 2010], and MNIST [Lecun et al., 1998].

The work by [Krizhevsky et al., 2012] "ImageNet Classification with Deep Convolutional Neural Networks" is a prime example of computer vision utilizing an extensive image data set successfully. The work was able to improve the top-5 test error rate in the *ImageNet Large Scale Visual Recognition Challenge* from 26.2% down to 15.3% after years of stagnation. This work also marks the renewed interest in deep learning approaches in computer vision. From 2012 onwards, improved error rates were reported yearly by deep learning approaches. Eventually, in 2015, the work of [He et al., 2015] is the first to surpass the reported human-level performance of 5.1% [Russakovsky et al., 2015] on this specific data set.

With respect to all the advancements in the field, most computer vision systems do not take into account *how* the data is captured. In fact, they usually act as a passive observer; humans handpick the input data, the camera has pre-determined settings and viewpoints, and often the domain is somewhat limited. Visual perception is more than just signal processing; it is an active process. Talos would not have been able to perform his duty being passive. He was active – wandering the island many times a day.

Visual perception is a problem of control of data acquisition and not necessarily one of signal processing, as argued by Bajcsy [Bajcsy, 1988, Bajcsy, 1985]. It is pointed out that the activity of perception is exploratory, probing, and searching. The analogy used is that

> "percepts do not simply fall onto sensors as rain falls onto the ground. We do not just see, we look."

This becomes even more clear when we think of a cluttered scene in which the object of interest is occluded by another object – We must move our head or even the occluding object in order to disocclude it.

Active computer vision, an area of computer vision, addresses exactly this. Essentially, active computer vision can be defined as a set of visual behaviours that use image interpretations to purposefully control intrinsic and extrinsic geometric parameters of the sensory apparatus to improve the quality of a particular vision task [Bajcsy, 1988, Aloimonos et al., 1988, Dickinson et al.,

1997, Crowley et al., 1992].

In this thesis, we go one step further as most of active computer vision presents pure engineering solutions. The work proposed here is novel in the sense that it aims to discover active *human visuospatial behaviours* – how humans behave as active observers when solving visuospatial tasks with the goal of using this knowledge to build robotic agents whose behavior is comparable to humans, as Talos was originally conceived. This is an important step towards building *human-like* visual systems, robots, etc., that aim to be a real-world assistant in a variety of settings. In Section 5.1.3 we define in detail what this means. In short, our effort seeks to go beyond correct input-output behaviour as proposed with the Turing Test and also includes human-like steps between input and output, the amount of data required, time to learn, error rates, as well as the type of errors.

The objective of this chapter is to discuss why human visual behaviours are fundamental to improving real-world active vision systems. Before doing so, it is important to present a definition of active vision, what the progress in developing active vision systems is and what remains unsolved. To accomplish this, we present a brief history of active vision systems, discuss some of the significant milestones, and present unresolved challenges.

1.2 Active Vision

Active computer vision and its necessary background are described in this section. In the process, we will present a definition of active computer vision, its connection to Marr's theory of vision, and provide a brief history of active vision systems.

1.2.1 Active Vision and Marr's Theory

The influence of Marr's work, especially [Marr, 1982, Marr and Nishihara, 1978], for the computer vision, human perception, and human cognition fields is undeniable. He was able to explain the processes that are relevant for vision in a more precise and concise format than many other authors in far larger volumes [Mertsching and Schmalz, 1999]. He provided a foundation to build models in vision and describes the relevant processes, starting with early visual processing to three-dimensional model representation and object recognition. Besides this, what makes his work distinguished, is

that he put the focus on the biological plausibility of his algorithms.

Marr views vision as a capability that can be entirely formalized as a pure information processing task and can be based on general theory. This implied a closed framework to deal with visual tasks and the existence of general solutions. Further, according to his methodology, in order to understand a perceptual process, he suggests three levels "at which an information-processing device must be understood before one can be said to have understood it completely" [Marr, 1982]:

- *Level I: Computational Theory.* "What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?"
- *Level II: Algorithms and Data Structures.* "How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?"
- *Level III: Hardware Implementation.* "How can the representation and algorithm be realized physically?"

If all three levels are understood, we can understand the visual perception process under examination. Some researchers include an additional level to address the stability [Horn and Weldon, 1987, Huang and Blonstein, 1985, Blostein and Huang, 1987, Adiv, 1989, Hummel, 1987, Bajcsy, 1988] and the complexity level [Tsotsos, 1987] of the system. [Aloimonos and Shulman, 1989] positions a *stability Analysis*-Level between *Level II* and *Level III* to enrich the Marr paradigm. The argument is that while developing the computational theory, noise contained in the input data is not taken into account and "we may get absurd results". This is the reason why a stability analysis has to be taken care of.

Besides the enthusiasm surrounding Marr's book [Marr, 1982], the limitations are apparent. He describes the human visual system as a well-defined, closed information processing system and visual perception is seen as a reconstruction process of the sensory input. This kind of problem is only solvable under well-defined conditions and is, therefore, an ill-posed problem. [Hadamard, 1902] believed that mathematical models of physical phenomena should have the following three properties:

1. A solution exists.

2. The solution is unique.

3. The solution's behaviour changes continuously with the initial conditions.

If one or more of these criteria are not satisfied, the problem is *ill-posed*. Almost every fundamental problem in computer vision is challenging because there does not exist a unique solution. [Aloimonos et al., 1988] shows that many ill-posed problems (shape from shading, shape from contour, shape from texture, structure from motion, and optic flow) can become well-posed when taking into account an active observer who is capable of gaining other visual information from different viewpoints.

Further, [Mertsching and Schmalz, 1999] points out that Marr's strict representation scheme prevented the introduction of continuous learning, adaption, and generalization. This is actually an interesting and intriguing omission in Marr's theory. A comment is provided in the afterword of the revised version of Marr's book about 30 years later by Tomaso Poggio:

> "I am sure that this omission would have been corrected had Marr had the time. [...] Of course, it is important to understand the computations and the representations used by the brain – this is the main objective of the book – but it is also important to understand how an individual organism, and in fact a whole species, learns and develops them from experience of the natural world. [...] I am not sure that Marr would agree, but I am tempted to add learning as the very top level of understanding, above the computational level. We need to understand not only what are the goals and the constraints of computation are but also how a child could learn it and what the role of nature and nurture is in its development. Only then may we be able to build intelligent machines that could learn to see – and think – without the need to be programmed to do it."

Taking all of this into account, his theory is essentially about passive vision, as he neglects the perceiver's behaviour. He addresses such drawbacks by writing: "all other facilities can be hung off a theory in which the main goal of vision was to derive a representation of shape". Marr saw vision independent from the observer and the particular visual task, hence, a general process. [Aloimonos, 1995] criticizes Marr by saying concerning his work that "Vision was studied in a disembodied

manner by concentrating mostly on stimuli, sensory surfaces, and the brain". Marr's approach was to analyze the entire image, while biological vision systems use selective representations about the world with respect to the task and context. [Bajcsy, 1985] describes the physical adjustments of our visual system in the course of looking, as "...our pupils adjust to the level of illumination, our eyes bring the world into sharp focus, our eyes converge or diverge, we move our heads or change our position to get a better view of something, and sometimes we even put on spectacles".

Simply put, general-purpose vision, as described by Marr, is restricted to certain, limited domains. Image data is just too rich, and there is too much to be known about the world for us to construct a task-independent description. [Aloimonos, 1993] gives an example that addresses the problem of a general-purpose vision theory with respect to the human brain: "...one of its special features is the fovea. Humans look at the world using a small window that they move around using a very elaborate gaze control system. If the resolution of the human eye were everywhere equal to its resolution near the optical axis, then humans would have a brain weighing approximately 30,000 pounds". Serious complexity issues of visual perception are addressed by [Tsotsos, 1987], who shows that the problem can be converted into a much simpler one by using several physical and biological constraints. The provided analysis argues in favour of the computational necessity of attentive visual processes. Later, [Tsotsos, 1989] shows formally that Marr's bottom-up view is intractable.

1.2.2 Defining Active Vision

The community of active computer vision addressed the different aspects of the problem and emphasized this by giving their approaches different characteristic terms. In other words, active computer vision is known in the literature under many different names. The most relevant versions with respect to this work will be described. All of these versions use visual sensory input as their data to purposefully control camera parameters, with the inclusion of feedback to gather data as needed. We will continue now to introduce the different terms, give a broad overview for each of them, and conclude with the definition of active computer vision as used in this work.

Active Vision is the most general term (and also the one used in this dissertation). The approaches termed *Active Vision* try to overcome the drawbacks of Marr's theory of vision by introducing an active observer. Examples are [Aloimonos et al., 1988, Wilkes, 1994, Dickinson

et al., 1997, Browatzki et al., 2012]. [Aloimonos and Shulman, 1989] describes it by stating, "an observer is called active when engaged in some activity whose purpose is to control the geometric parameters of the sensory apparatus. The purpose of the activity is to manipulate the constraints underlying the observed phenomena in order to improve the quality of the perceptual result". The overall idea is to optimize the quality and quantity of the visual data as needed and to break down the complexity of the task [Landy et al., 2012, Swain and Stricker, 1993, Vieville, 2012].

Purposive Vision, also known as *Qualitative Vision*, is closely connected to *Active Vision*. It stresses the point that the *purpose* is the driving motivation to interact with the environment and to determine the efforts to realize the visual task [Aloimonos, 1993, Aloimonos, 1994]. Therefore, the important bit is to ask what vision will be used for. It is described that "it depends on the tasks that the system has to carry out, i.e., on its purpose. Purposive vision does not consider vision in isolation, but as part of a complex system that interacts in specific ways with the world".

Animate Vision is introduced by [Ballard, 1991] and uses the human visual system as a rich source for technical inspirations. In [Ballard and Brown, 1992], he writes "...researchers...seek to develop practical, deployable vision systems using two principles: the active, behavioural approach, and task-oriented techniques that link perception and action".

Dynamic Vision was introduced by Ernst Dickmanns as an extension to computer vision to deal with scenes with several moving objects, including the camera itself [Dickmanns and Graefe, 1988, Dickmanns, 1997, Dickmanns, 2007]. No assumptions were made whether the background was stationary or not. This approach was developed mainly for the field of autonomous driving and was able to compensate not only for intended motion but also for unavoidable perturbations, such as pitching and rolling motion from cars on uneven surfaces or for aircraft from wind gusts. Taking these conditions into account, it is proposed to join inertial and visual sensing as it provides the advantages necessary to overcome the drawbacks, as mentioned earlier.

Active Perception stresses the combination of modelling and control strategies for perception [Bajcsy, 1988, Bajcsy, 1996, Bajcsy and Campos, 1992, Bajcsy et al., 2017]. Active perception "...can be stated as a problem of controlling strategies applied to the data acquisition process which will depend on the current state of the data interpretation and the goal or the task of the process" and a closed feedback loop is necessary "to define and measure parameters and errors from the scene which in turn can be fed back to control the data acquisition process" [Bajcsy, 1988].

The Ecological Approach to Visual Perception by [Gibson, 1979] emphasizes the way an active observer picks up information from the environment. The central points of J. J. Gibson's approach is that all the information needed to form perception is available in the environment, and crucial information for perception is information that remains largely invariant as an observer moves through the environment. For him, perception and action cannot be separated. [Goldstein, 1981] concludes that "Gibson's concern with the characteristics of the information responsible for perception led him to emphasize the fact that real-life perception involves not a stationary observer fixating on a small light in a laboratory, but, rather, an active observer who is constantly moving his or their eyes, head and body relative to the environment."

Besides these five terms, active computer vision can be found in the literature as *Behavioural Vision* [Ballard, 1989, Livingstone and Spacek, 1996], *Utilitarian Vision* [Aloimonos, 1992, Rivlin et al., 1992], *Embodied Vision* [Arsenio, 2003, Zhu et al., 2021, Aubret et al., 2022] (and more).

From here on, we will use the term *active vision* and follow the definition of [Bajcsy et al., 2017]:

> "An agent (active vision system) is an active perceiver if it knows *why* it wishes to sense, and then chooses *what* to perceive, and determines *how*, *when* and *where* to achieve that perception."

[Bajcsy et al., 2017] describes this as the active-pentuple that defines an active vision system:

$$AP = (why, what, how, when, where) \tag{1.1}$$

Naturally, resource constraints play an important role not only because of computer power, power efficiency, specifically for mobile active vision systems and memory capacity, but also because the number of sensors and other physical components of an active vision system are limited as well. These constraints become a significant factor in determining the viability of the constructed vision system [Andreopoulos and Tsotsos, 2013, Bajcsy et al., 2017]. Thus, choices must be made.

To conclude this section, we want to illustrate the difference between passive and active vision with Figure 1.3 adapted from [Bajcsy et al., 2017]. The figure shows high-level processing pipelines of passive computer vision (top) and active computer vision (bottom). The diagram presents the key

difference between active and passive vision; Passive vision is a feed-forward, information-processing approach, following mostly the Marr paradigm as described earlier, whereas active vision systems attempt to deal with the perceptual-motor loop and include the *Why* constituent with at least one other constituent, whereas, as described by [Bajcsy et al., 2017] a complete active agent would at least include one component from each element of the active pentuple.

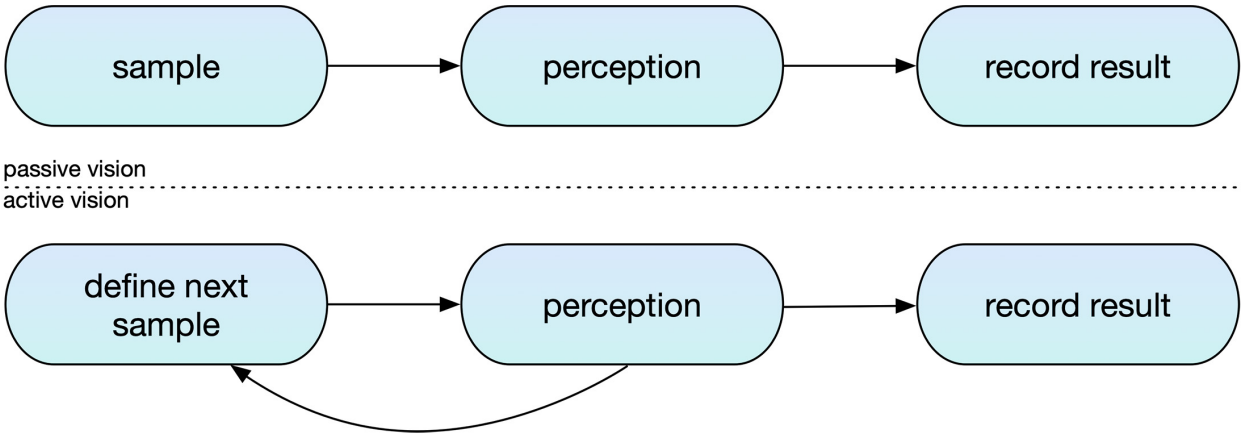


Figure 1.3: Processing pipelines of passive (top) and active (bottom) vision. Adapted from [Bajcsy et al., 2017].

1.2.3 Literature Review

While tremendous advances have been shown in traditional, single view computer vision over the last few decades, and especially over the last 10 years, it is faced with inherent problems:

1. *Impossibility to invert projection and fragility of three-dimensional inference* – Unless we make restrictive assumptions, it is impossible to recover a three-dimensional representation from its two-dimensional projection on an image [Palmer, 1999].

2. *Occlusion* – Visual features, necessary for the visual task at hand, might be self-occluded or occluded by other visual features [Marr, 1982].

3. *Detectability* – Visual features might be missing or undetectable due to mis-matched settings of the camera parameters, illumination conditions, and incorrect camera placement [Andreopoulos and Tsotsos, 2012].

4. *View degeneracies* – Wrong feature interpretations due to view degeneracies that are caused by accidental alignments [Dickinson et al., 1999].

From the above four problems, one can easily see that the processing pipeline of passive vision can be negatively influenced. Various methods are proposed to address these problems.

Satisfaction	**Constraint**
G1	Visibility
G2	Viewing angle
G3	Field of view
G4	Resolution constraint
G5	In-focus or viewing distance
G6	Overlap
G7	Occlusion
G8	Image contrast
G9	Kinematic reachability of sensor pose

Table 1.1: Sensor placement constraints [Chen and Li, 2004].

Active vision, and therefore sensor planning, was in the early days mainly focused on the analysis of placement constraints, such as resolution, focus, field of view, visibility, and conditions for light source placement. The goal was to place a viewpoint in an acceptable space and to satisfy a number of constraints. A list of constraints is provided in Table 1.1 and can be summarized as geometrical placement (G1, G2, G3), optical (G3, G5, G8), reconstructive (G4, G6), and environmental (G9) [Chen and Li, 2004]. Common methods and solutions regarding the view-pose determination and sensor parameter setting can be roughly categorized into the following seven main groups [Chen et al., 2011]:

1. *Formulation of Constraints*

   Approaches in this category focus on that the intended observation needs to satisfy a number of constraints [Chen et al., 2011] (see Table 1.1). Constraints are mostly used in model-based vision tasks [Trucco et al., 1997], and include inspection, assembly, recognition, and visual search [Tarabanis et al., 1995]. However, approaches exist for non-model based tasks as well [Chen and Li, 2005, Chen et al., 2008]. [Tarabanis et al., 1995] for instance provides an approach that formulates the probing strategy as a function minimization problem. The function is the weighted sum of several component criteria. Other constraint formulation

methods include [Chu and Chung, 2002, Tarabanis et al., 1994, Tarabanis et al., 1996].

2. *Expectation*

   Local surface features together with expected model parameters are frequently used in view-pose determination [Flandin and Chaumette, 2001]. Specifically, [Jonnalagadda et al., 2003], proposed a strategy to select view-poses in four steps: (1) local surface feature extraction, (2) shape classification, (3) view-pose selection, and (4)global reconstruction. Simple geometric primitives are assembled from two-dimensional and three-dimensional surface features. The primitives are then classified into shapes. In turn, the shapes are used to hypothesize the global shape of the object and plan the next view-pose. Other expectation approaches include [Fiore et al., 2008, Ellenrieder et al., 2005, Kutulakos and Dyer, 1994].

3. *Multi-agent approach*

   These approaches employ multiple agents that collect data differently from the same environment. For example, [Mostofi, 2011] proposed a framework to build a map using multiple agents with a small number of measurements. [Bakhtari et al., 2006, Bakhtari and Benhabib, 2007] propose a method for dynamic coordinated selection and positioning of active-vision cameras to simultaneously surveil multiple objects as they move through a cluttered environment with unknown trajectories. The goal of the system is to adjust the camera poses dynamically in order to maximize object visibility and acquire images from preferred viewing angles. Other multi-agent approaches include [Naish et al., 2003, Hodge and Kamel, 2003, Lim et al., 2007, Suppa and Hirzinger, 2007, Flandin and Chaumette, 2002]

4. *Statistical approaches*

   Statistics, probability, Kalman filters, and associative Markov networks have been widely used in the field of active object recognition [Wheeler and Ikeuchi, 1995, Dickinson et al., 1997, Roy et al., 2000, Caglioti, 2001], grasping [Motai and Kosaka, 2008], and modeling [Triebel and Burgard, 2008]. More precisely, in sensor planning for object search, a robot action is defined by viewpoint, a viewing direction, a field of view, and the application of the recognition algorithm. [Ye and Tsotsos, 1999] formulates this as an optimization problem with the goal to maximize the probability of detecting the target object within a cost limit. As shown by [Ye and Tsotsos, 1999], this problem is $NP-hard$. To efficiently determine the

sensing actions over time, selection policies were proposed. Such policies were further studied in [Shubina and Tsotsos, 2010] and [Rasouli and Tsotsos, 2014, Rasouli and Tsotsos, 2015]. The next action is selected based on the likelihood of detection and the cost of the action. If the detection is unlikely, the agent is moved to another position where the probability of detection is highest. Another example, that addresses the illumination condition of the detectability is proposed by [Vázquez, 2007]. Here, the Shannon entropy is applied to the problem of automatic selection of light positions in order to maximize visual information recovery.

5. *Learning Methods*

   This category spans across multiple learning methods, such as evolutionary algorithms, fuzzy inference, neural networks, rule-based planning, reinforcement learning, and expert systems. For instance, [Kwok et al., 2006] employed an evolutionary approach in the context of SLAM (Simultaneous Localization and Mapping). In another example, [Deinzer et al., 2009] used an unsupervised reinforcement learning algorithm for active view-pose selection for object recognition. [Yang et al., 2019] present an embodied version of Mask R-CNN [He et al., 2017] for active object recognition. Here, Mask R-CNN is used to recognize objects of interest visually and reinforcement learning, specifically REINFORCE [Sutton et al., 1998] is used to learn how to move given the output of Mask R-CNN. This approach achieved higher accuracies across all evaluations than the passive baseline. An example illustration is shown in Figure 1.4.

6. *Dynamic configuration*

   In several active vision systems, the robot moves from one place to another to perform a multi-view task. A traditional vision sensor with a fixed configuration is often inadequate [Chen et al., 2011]. For instance, an active approach for coarse-to-fine image acquisition is proposed by [Das and Ahuja, 1996]. The following steps are suggested to aim cameras in a different direction and to fixate at different objects. (1) A new fixation point is selected from non-fixated, low-resolution scene parts. (2) A re-fixation of the cameras is initiated. This is an iterative process – as the camera is reconfigured, the fixation point ideally gradually deblurrs which allows for a more precise camera configuration. (3) Finally, the focus settings

are completed, allowing for improved sensing.

7. *Active Lighting*

   These approaches aim to optimize the light position to achieve adequate illumination, mathematically through the light path, such as surface absorption, diffused reflectance, specular reflectance, and image irradiance [Chen et al., 2011]. One such approach is provided by [Eltoft and DeFigueiredo, 1995]. It is found that illumination control can be used to enhance image features (points, edges, shading patterns, ...) which can provide essential cues for the interpretation of an image. As described earlier, [Vázquez, 2007] is also an approach that falls under this category.

The above methods might be used independently, or as hybrids, in applications and tasks such as Purposive sensing (e.g. robot understanding with a focus on efficiency and accuracy), Mobile Robotics (for instance, SLAM, Navigation, Exploration), Robotic Manipulations, Object modelling, Site modelling, Surveillance (for example Tracking, Search), and more.

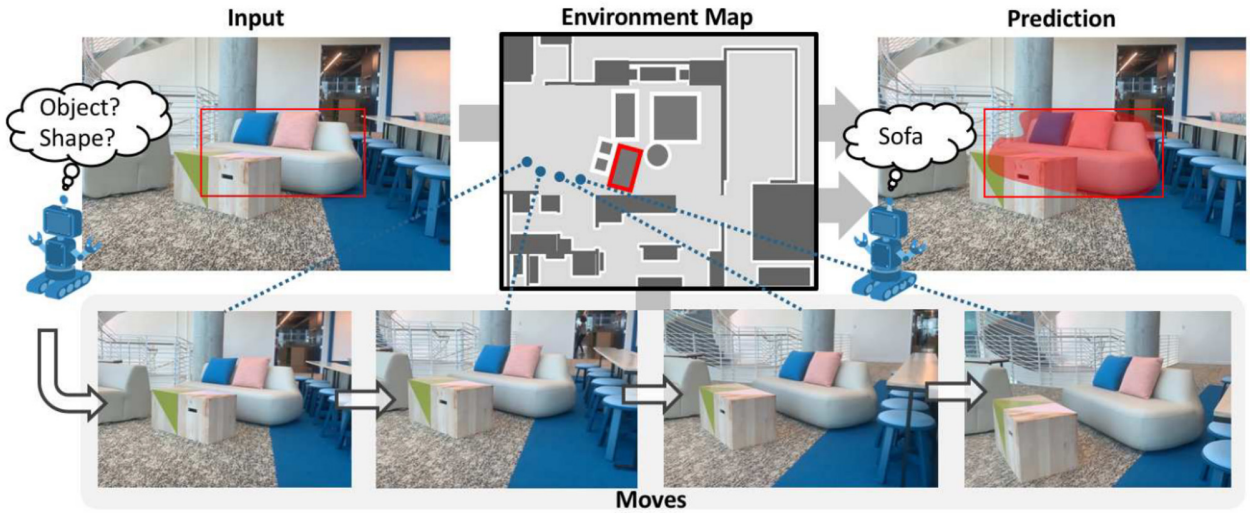


Figure 1.4: Embodied Recognition. The agent starts close to an occluded target object and needs to move around to aggregate information to increase the recognition quality. Source: [Yang et al., 2019].

In general, gaining information from image data is a hard problem [Hanson, 1978, Tsotsos, 1989, Tsotsos, 1992, Bajcsy et al., 2017]. However, by endowing a vision system with the ability to be active and choose the next observation, complexities of vision tasks can be simplified [Aloimonos et al., 1988], such as visual object recognition. An early example of active object recognition was

given by [Wilkes and Tsotsos, 1993].

[Wilkes and Tsotsos, 1993] exploits the mobility of the sensing unit by using the image data to direct the camera to a particular viewpoint. A sketch is shown in Figure 1.5. The authors define *special viewpoints* of objects that yield robust and reliable views of the objects that reduce the task to a two-dimensional pattern recognition problem. Hence, actively seeking a *special viewpoint* is preferred. This is just one example that shows that active behaviours simplify the already hard task at hand.

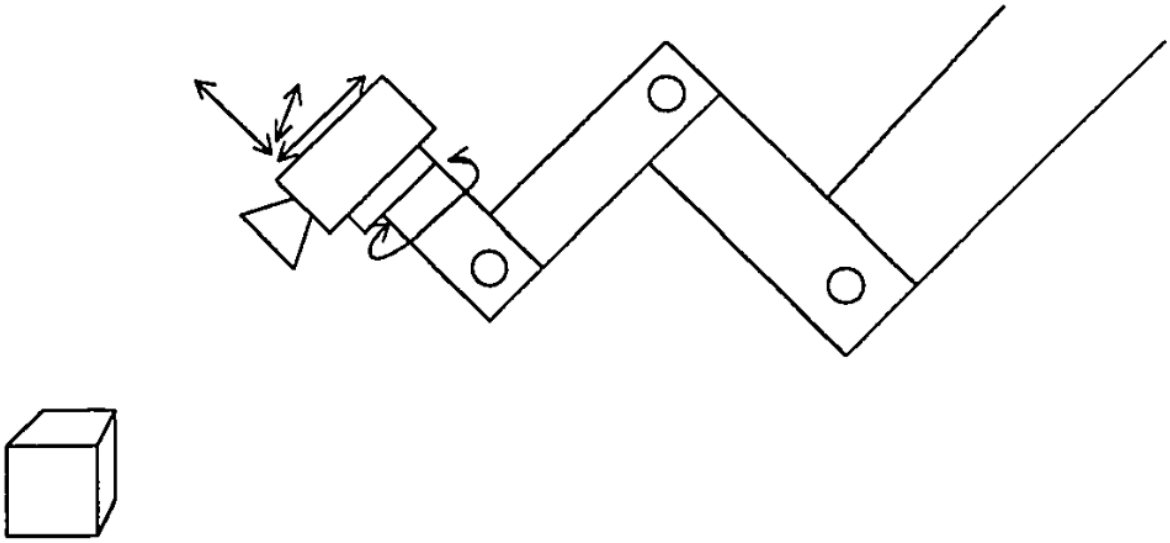


Figure 1.5: Setup for 3D active object recognition: A viewpoint-controllable 2D sensor. Source: [Wilkes and Tsotsos, 1993].

Most of human vision is active by nature; change of viewpoint, saccades, smooth pursuit, vergence, and other eye movements are actions humans experience continuously. Humans are active observers in their everyday lives; they explore, search, select what and how to look [Bajcsy, 1985, Bajcsy, 1988, Bajcsy et al., 2017]. For a review see [Findlay et al., 2003]. Nonetheless, how exactly active observation occurs in humans so that it can inform the design of active computer vision systems is an open problem [Aloimonos et al., 1988, Crowley et al., 1992, Dickinson et al., 1997].

This thesis addresses exactly this; how does active observation manifest in humans and how can this knowledge inform computational solutions for active vision systems. No detailed data exist to

investigate this, until now.

The breadth of tasks that humans perform actively is enormous (for instance, the DARPA Grand Robotics Challenge of 2015 gives several tasks, all of which remain open challenges). We sought to find commonalities among tasks in order to explore very basic components of a wide variety of tasks. One sub-task that seems common to almost all everyday activities is the ability to compare one thing to another. Whether in memory or "live", comparison seems to be a fundamental behaviour. As such, comparisons have been studied by cognitive science extensively in humans and animals. One example study that has been very influential is due to [Shepard and Metzler, 1971]. They looked at rotated configurations of blocks and asked subjects to determine if two configurations were the same or different. This was a passive, two-dimensional task. Such a task seemed sufficiently basic if we extended it to three dimensions and an active format.

In order to do so, we introduce a challenging set of unfamiliar objects of the same sort as those of Shepard and Metzler [Shepard and Metzler, 1971], and hence fall into the blocks world realm of [Roberts, 1963], but in three-dimensions and with controlled and measurable complexity. We use the novel stimuli to test and record human participants performing a version of the well-studied same-different task, but in three-dimensions and allowing for active observation. We have conducted a large scale experiment with dozens of randomly sampled participants and have recorded thousands of fixations in an controlled environment with different experimental variables. We use the recorded data to analyze and disect fixation sequences to understand exactly *how* the participants solve this task.

Our approach chooses a different angle to this problem than most other active perception approaches proposed, for instance, by [de Melo et al., 2021] and all of the reviewed work presented before. [de Melo et al., 2021], in more detail, proposes that computational perception will advance by new power in simulators and multimodal data sets – in other words, by more data alone. We, in comparison, take a look at precisely *how* humans solve visual tasks as they are remarkably good at this, as well as *human-like* behaviours are desirable for many real-world robotic systems (Section 5.1.3 provides a detailed discussion).

1.3 Significance

Active vision is a core ability we, as humans, use many times a day. The findings of this thesis have an impact on any robotic vision system whose role it is to be a real assistant at home, manufacturing, service or medical setting.

Eventually, the ability to identify human-like active behaviours is essential for developing any robotic system that interacts with and whose behaviour needs to be understood by humans.

PR2 by Willow Garage has been programmed to fold laundry [Srivastava et al., 2015], *gita* helps to carry heavy groceries [Lynn and Olsen, 2018], *ROBEAR* is designed for elderly care, able to lift a person [Davies, 2016], *HRP-5P* is able to assists on a construction side by carrying dry-wall [Kaneko et al., 2019], *spot* loads and unloads a dishwasher [Norman, 2005], *Motoman* does preparatory work for cooking meals [Kusuda, 2010], and *Nao* [Gouaillier et al., 2009] picks up toys and places them in a bin. Figure 1.6 shows them in action. This is just the beginning of an ongoing list of robotics applications with the need for human-like active behaviours to support humans in their home setting; clean the kitchen, clean toilets and bathtubs, pick up objects and move them to the right location, unload groceries from the car, assemble furniture, or find your lost keys.

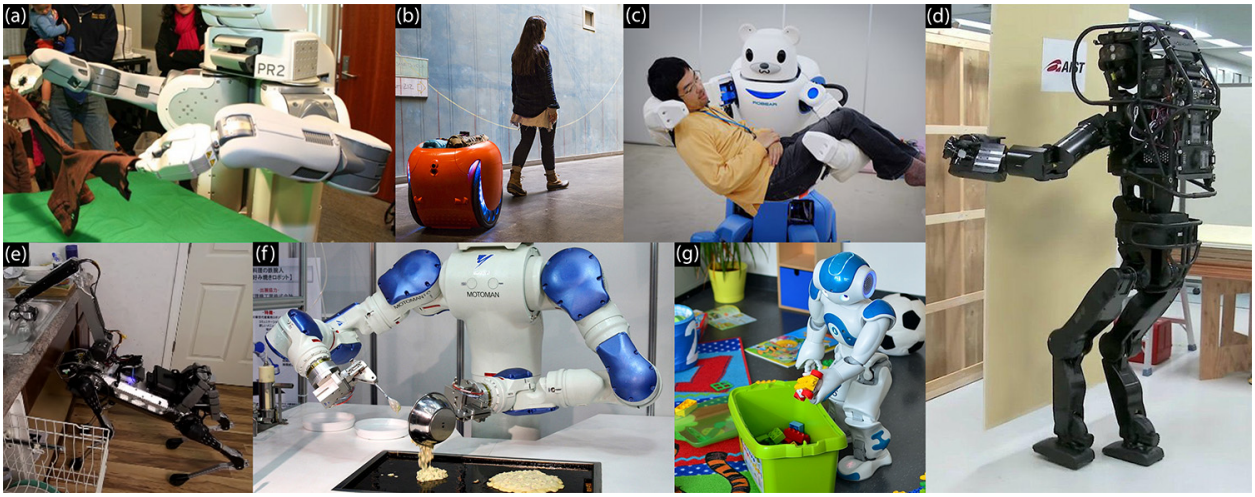


Figure 1.6: A collage of existing agents colaborating with humans. (a) PR2 folding laundry, (b) mobile-carrier robot gita, (c) robot for elderly care ROBEAR, (d) dry-wall carrying robot AIST, (e) dishwasher loading robot spot, (f) cooking robot Motoman, (g) Nao cleans up toys. Credits: (a) `http://www.eecs.berkeley.edu/~pabbeel/`, (b) `https://www.piaggiofastforward.com/gita`, (c) `https://www.theverge.com/2015/4/28/8507049/robear-robot-bear-japan-elderly`, (d) `https://www.aist.go.jp/index_en.html`, (e) `https://tinyurl.com/y2taeckq`, (f) `https://tinyurl.com/y28dpdf5`, (g) `http://www.squirrel-project.eu/objectives.html`, accessed: 01 January, 2020.

In conclusion, everyday behaviours for computational agents, such as recognizing objects, rely on sequences of decisions and physical actions. As shown by [Ye and Tsotsos, 1999], the problem of selecting a series of actions to accomplish a goal by taking into account resource constraints is $NP - hard$. Notwithstanding, humans are remarkably capable of doing this [Bajcsy et al., 2017].

1.4 Organization of Dissertation

The remainder of this document is organized as follows, Chapter 2 introduces a variation of the well-studied Shepard & Metzler same-different task, extended to three-dimensional objects in the real world and permitting subjects to be active observers. We provide a brief summary of human visuospatial abilities, review instantiations of the traditional same-different task and introduce our three-dimensional version of the same-different task – *Three-Dimensional Same-Different Task for Active Observers*. For this, we propose a novel set of three-dimensional objects to push the limits of visuospatial observations, both for humans and machines. These objects are sufficiently complex to effectively act as probes exploring the range of active human observation during visual problem-solving. We describe in detail the characteristics and how we designed the objects. Additionally, we also establish an image data set based on the objects with rich annotations and define a self-occlusion metric as self-occlusion plays an influential role for visual tasks in general. In order to test this, we chose a number of modern deep-learning classification systems and try to learn the objects. Having shown that the recognition of these objects is non-trivial, we next turn to how human subjects observe these objects. With *PESAO* we present a novel psychophysical experimental setup for active observers that enables the detailed collection of data from active observers attempting to solve visuospatial problems involving physical three-dimensional objects and environments. We cover the necessary background of why we decided to build *PESAO*, including a review of existing systems, as well as detailed descriptions of the hardware used and software created. Lastly, we provide all necessary information on how we run the experiment to allow us to reproduce and adapt the experiment.

In Chapter 3 we investigate human visual behaviours on the example of the three-dimensional same-different task using our novel objects and record head and gaze using *PESAO*. We have recorded dozens of subjects totalling hundreds of trials in a detail that has not been done until

now. The chapter provides further details on the conducted experiments for reproducibility and presents a number of different performance metrics such as accuracy, response time, the number of fixations and more. In addition, we present a complexity level analysis.

An analysis of our collected data is provided in Chapter 4. We mined fixation sequences, propose how visuospatial strategies are composed, and describe their connection to cognitive programs [Tsotsos and Kruijne, 2014].

This raises the question of how our findings compare to strategies used by modern machine learning methods. This is addressed in Chapter 5. We present our approach to learning the same-different task with a modern machine learning method. An overview of modern machine learning methods is provided, as well as an explanation of why reinforcement learning, in our opinion, is best suited for the same-different task. We continue with an introduction to reinforcement learning and present our efforts to learn the task, including the implementation of different virtual three-dimensional same-different tasks. We conclude the chapter with an interpretation of the results, including suggestions for the development of future machine learning methods.

Lastly, Chapter 6 will conclude this document and provide future directions for the presented work.

Chapter 2

Three-Dimensional Same-Different Task for Active Observers

Sections in this chapter have been published previously in the following:

Section 2.2: Markus D. Solbach and John K. Tsotsos "Blocks World Revisited: The Effect of Self-Occlusion on Classification by Convolutional Neural Networks", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (Workshop: Real-World Computer Vision from Inputs with Limited Quality (RLQ))*, 186, [2021]

Section 2.3: Markus D. Solbach and John K. Tsotsos "Tracking Active Observers in 3D Visuo-Cognitive Tasks", in *ACM Symposium on Eye Tracking Research and Applications*, 1-3, [2021]

Section 2.4: Markus D. Solbach and John K. Tsotsos "Active Observers in a 3D World: The 3D Same-Different Task", in *Journal of Vision* (Abstract), Vol. 20, No. 11, [2020]

2.1 Introduction

In this chapter, we propose a three-dimensional version of the well-studied same-different task for active observers. As mentioned previously, the task and our particular set of stimuli permit us to probe and examine the space of active human observation during visual problem-solving. First, we will provide background on human visuospatial abilities in general, explain why we chose the same-different task as our testbed and what are our goals for this experiment.

Visuospatial abilities describe the capacity to understand, reason about, and remember the spatial relations among visual stimuli [Donnon et al., 2005]. The breadth of visuospatial tasks that are performed by humans daily are stunning [Carroll, 1993]. One example is the famous Rubik's cube (Figure 2.1, where the goal is to sort the six colours dependent on the six sides of the cube such that one side only shows one colour. One is allowed to rotate the elements around three spatial axes, which makes it a three-dimensional puzzle, hence requiring visuospatial capabilities, for example three-dimensional mental rotation.

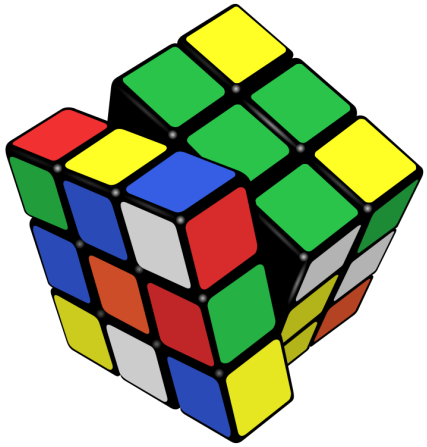


Figure 2.1: A Rubik's cube: A popular three-dimensional puzzle that involves visuospatial abilities in order to solve. Source: `https://commons.wikimedia.org/wiki/File:Rubik%27s_cube.svg`

While the list of visuospatial tasks is beyond the scope of this work, we wish to provide a brief overview and necessary background to understand the task we have chosen for this dissertation.

2.1.1 Examples of Human Visuospatial Abilities

A great resource of cognitive abilities of humans is provided by [Carroll, 1993], especially Chapter 8 *Abilities in the Domain of Visual Perception.* Here, we will present three such visuospatial abilities

in the domain of visual perception; Spatial Visualization, Speeded Rotation, and Visuospatial Perceptual Speed. Figure 2.2 provides examples for each of them.

The ability of spatial visualization (Figure 2.2 A) describes the process of apprehending, encoding, and mentally manipulating spatial forms. Examples are paper folding or spatial relations. In the version shown in Figure 2.2, a model is provided (top), and the task is to determine which of the four objects (bottom) can be created by folding along the dashed line[1].

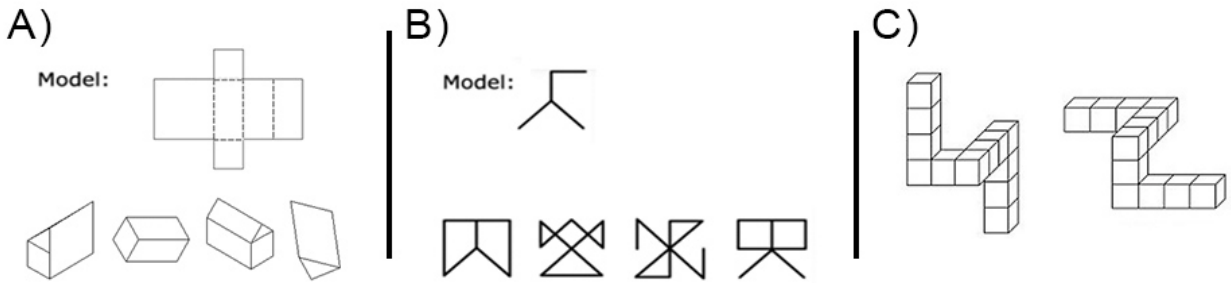


Figure 2.2: Three examples of visuospatial abilities: A) Spatial Visualization, B) Visuospatial Perceptual Speed, C) Speeded Rotation. Adapted from [Wanzel et al., 2003, Luursema et al., 2012].

Visuospatial Perceptual Speed (Figure 2.2 B) describes, for example, the task of quickly deciding whether a simple target pattern is present in a more complex pattern. The example in Figure 2.2 shows the target pattern above and the four possible patterns to compare against below[2].

Lastly, Speeded Rotation requires mental transformations of three-dimensional objects projected two-dimensionally. An example is shown in Figure 2.2 C. This task presents two objects, perhaps in different orientations, and the goal is to tell whether they are the same or different[3]. This task represents the root of motivation for our task which is described next.

2.1.2 The Same-Different Task

The Same-Different task, also called *comparison task* or *matching task*, generally speaking is used to explore the concepts of "sameness" and "difference" [Harding, 2018]. The task is to judge as accurately and rapidly as possible whether two presented stimuli are the "same" or "different."

The classic instance of the same-different task is widely known from the work of [Shepard and Metzler, 1971]. Figure 2.3 shows three examples with different rotations of the objects; A), the

[1] A) Answer: Second object
[2] B) Answer: First and last object
[3] C) Answer: Same

objects are a "same" pair and differ by 80° rotation in the picture plane; B), the objects are a "same" pair and differ by 80° in-depth; C), a "different" pair, which cannot be brought into congruence by any rotation. Note how self-occlusion plays a large role for three-dimensional objects, especially if only one viewpoint is provided as here.

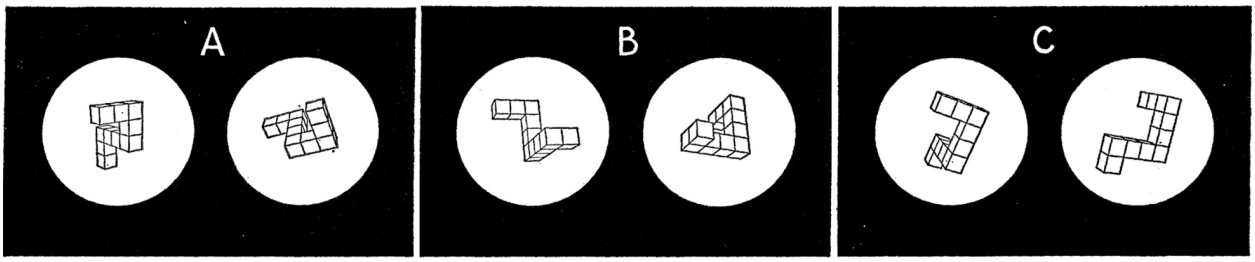


Figure 2.3: Examples of pairs of perspective line drawings used for the same-different task. Note how self-occlusion plays a large role for three-dimensional objects, especially if only one viewpoint is provided as here. Adapted from [Shepard and Metzler, 1971].

In the original study, different rotation angles, rotations with respect to picture-plane and depth were taken into account. In total, ten rotations were investigated from 0° to 180° in 20° steps. It is found that the reaction time increases linearly with the angular difference in portrayed orientation and to be no longer for rotations in-depth than for a rotation in the picture plane. Interestingly, humans are remarkably good at this task; the study reported that, on average, only 3.2% of the responses were incorrect (ranging from 0.6% to 5.7% for individual subjects).

The same-different task is not only bound to psychology [Shepard and Metzler, 1971, Brown and Austin, 2021, Farell, 1985, Davis and Goldwater, 2021], versions of this task are used in other research fields such as cognitive science [Martinho and Kacelnik, 2016, Van Opstal, 2021], health science [Katz and Wright, 2021], neuroscience [Basile et al., 2015] and others. Variants of this task include the comparison of letters (for example, [Bamber, 1969, Nickerson, 1965, Bamber, 1972, Bamber and Paine, 1973, Krueger, 1973, Taylor, 1976]), numbers (for instance, [Snodgrass, 1972, Silverman and Goldberg, 1975, Van Opstal and Verguts, 2011]), words (for example [Farell, 1977, Well et al., 1975]), faces (for instance, [Tversky, 1969, Megreya and Burton, 2006]), abstract patterns (for example, [Dyer, 1973, Nickerson and Pew, 1973, Link and Tindall, 1971, Nickerson, 1967, Egeth, 1966, Snodgrass, 1972]), motion direction (for instance, [Petrov, 2009]), and tones (for example, [Bindra et al., 1965, Bindra et al., 1968, Nickerson, 1969]).

Furthermore, in the field of computer vision, the same-different task is getting increasingly

popular as well [Koch, 2015, Kim et al., 2018, Harding, 2018, Han and Charles, 2019, Funke et al., 2021]. The concept of telling apart "same" and "different" as described earlier is a fundamental capability of humans. So, it is not surprising that efforts are undertaken to model this capability with algorithms.

For instance, [Kim et al., 2018, Stabinger et al., 2016] show that deep learning models fail to learn the particular forms of same-different. Specifically, [Kim et al., 2018] shows that the stimulus variability of this task makes rote memorization difficult. Figure 2.4 (left) shows a photo of a person playing the flute. The authors state that the flute can be "confidently classified," however, the right side, a same-different example stimulus, taken from the SRVT dataset [Fleuret et al., 2011], strains deep learning networks. In more detail, the best-performing CNN model was not able to get above chance from one million training examples. It is reported are that visual relation quickly exceeds the representational capacity of feedforward networks. Feature templates for single objects seem to be a significantly easier problem for modern deep networks; learning feature networks for *arrangements* of objects becomes intractable due to the combinatorial explosion of the needed number of templates [Kim et al., 2018]. Further, it is pointed out that it has been long acknowledged by cognitive scientists [Marcus, 2003, Fodor and Pylyshyn, 1988] that notions of "sameness" and stimuli with a combinatorial structure are difficult to model with feedforward networks.

Figure 2.4: Modern vision algorithms can confidently classify that the image on the left contains a flute. The same algorithms, however, have problems to learn the concept of "sameness". An example image is shown in the right [Ricci et al., 2018]. Adapted from [Ricci et al., 2018].

In conclusion, the same-different task is a well-studied capability in many different research fields. However, all instantiations of this task with respect to human observers are in two dimensions and do not include the ability to observe the stimulus from a different viewpoint – meaning that observers are passive.

2.1.3 Goals of this Experiment

We propose a three-dimensional version of this task which also allows for active observation – *Three-Dimensional Same-Different Task for Active Observers.* This experiment is designed to investigate human visuospatial abilities in the real world with as few constraints as possible to allow for natural problem-solving.

The goals of this experimental design can be summarised as:

- Creation of a set of objects that are/have
  - Three-Dimensional
  - Novel and unfamiliar
  - Known complexity
  - Challenging
  - Self-Occlusion
  - Common-Coordinate system
- An experimental set up that is/allows for
  - Unrestricted movement of the subject
  - Precise tracking of gaze and head motion
  - Controlled environment
  - Random subject sampling
- An experimental design that allows for
  - A true three-dimensional version of the same-different task
  - Different Experimental Variables
  - Reproduceability

In the next section, we will introduce a novel set of real, unfamiliar, three-dimensional objects with known complexity levels that allow for a systematic analysis of the *Three-Dimensional Same-Different Task for Active Observers.*

2.2 A Novel Set of Objects: Blocks-World Revisited

We propose a novel three-dimensional blocks world set of objects that focuses on the geometric shape of three-dimensional objects and their omnipresent self-occlusion. It is important that objects themselves be intrinsically difficult. For instance, there is little difficulty in asking if an apple and sledgehammer are the same objects – they are clearly not. The task becomes trivial if a single glance suffices to recognize the objects. We need objects which are unfamiliar and require active work to characterize. One complicating object characteristic is self-occlusion, a characteristic that most non-convex objects will possess as viewpoint varies. Here, we define a straightforward yet precise self-occlusion measure.

Further, to evaluate if the objects can be easily learned by state-of-the-art classification deep neural networks, we created a detailed image data set and trained modern networks. Even though remarkable progress has been seen in object classification over the years, self-occlusion still presents significant challenges.

TEOS is a dataset with 48 three-dimensional objects, divided into two subsets of 36 and 12 objects. We provide 768 uniformly sampled views of each object, their mask, object and camera position, orientation, amount of self-occlusion, as well as the CAD model of each object. Figure 2.5 shows an example object from three random viewpoints. Furthermore, we present baseline evaluations with five well-known classification deep neural networks and show that *TEOS* poses a challenge for all of them. The dataset, as well as the pre-trained models, are made publicly available for the scientific community under `https://data.nvision.eecs.yorku.ca/TEOS`.

2.2.1 Background

Over most of the last decade, computer vision was pushed by efforts put into deep learning. The exact advent of this deep learning-dominated era is often dated to the ImageNet challenge [Russakovsky et al., 2015] in 2012. Since then, the performance of models on various tasks has been improving at unparalleled speed; for instance, image classification on the ImageNet dataset surpassed the reported human-level performance in 2015 [He et al., 2015]. Two of the enablers for the recent successes are faster computers, specifically graphic processors, and the availability of large-scale and often well-curated data sets to learn from.

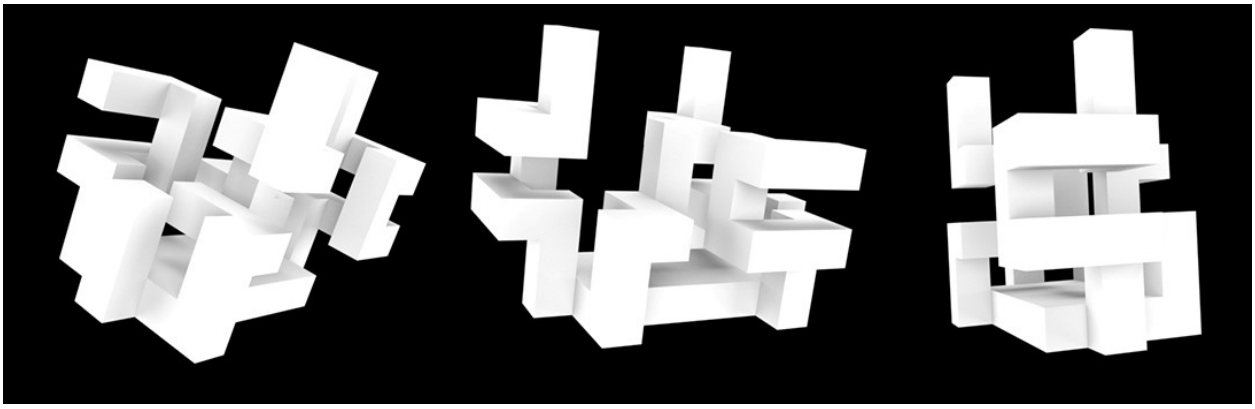


Figure 2.5: An example object of the proposed *TEOS* set of objects captured from three random viewpoints. Note that different views reveal but also hide different details of the object, which is due to self-occlusion.

The deep learning paradigm is ubiquitous, and, with it, the need for data with specific statistics to work in certain domains. [Kuznetsova et al., 2020] goes as far as saying that "Data is playing an especially critical role in enabling computers to interpret images as compositions of objects, an achievement that humans can do effortlessly while it has been elusive for machines so far."

Many domains exist in which one would like machines to perform visual tasks [Carroll, 1993]. One of these is object classification, which is defined as whether a particular item is present in the stimulus [Dickinson et al., 2009].

Object classification is an essential capability of humans, as well as for any robotic system whose goal is to be a real-world assistant; in a factory, hospital, or at home, just to name a few. Even though very successful in many domains, deep learning methods are challenged with occlusion [Koporec and Pers, 2019], which is inevitable in real-world scenarios. Here, we go a step further and show that deep learning methods are also challenged by the self-occlusion of objects.

The problem of understanding the three-dimensional structure from a two-dimensional description, for instance, a line drawing, was first put forward independently by [Huffman, 1971] and [Clowes, 1971], and they both showed that the necessary critical condition for a line drawing to represent an actual arrangement of polyhedral objects was label ability – that the lines and vertices could be be unambiguously labeled as being of a particular type.

As the human brain is very efficient at reconstructing a scene's three-dimensional structure from a single image with no texture, colour or shading, efforts have been concentrated on computational

complexity issues; one might think an efficient solution exists (e.g. polynomial-time). [Kirousis and Papadimitriou, 1988], however, proved that this problem is NP-Complete, also for simple cases like trihedral, solid scenes. To further research in this field, [Parodi et al., 1998] proposed a method to generate random instances of line drawings with useful distribution to investigate questions related to the complexity of understanding images of polyhedral scenes.

With the increasing successes, contemporary computer vision approaches show a trend away from artificial problems and provide solutions to real-world problems, already deployed in many domains [Andreopoulos and Tsotsos, 2013, Voulodimos et al., 2018], for example, optical character recognition, industrial inspection systems, medical imaging, and biometrics. While toy-domains are essential demonstration vehicles even in the deep learning era [Dosovitskiy et al., 2015, Ilg et al., 2018, Jalal et al., 2019, Johnson et al., 2017, Mayer et al., 2016], a disparagement of artificial domains can be seen [Slaney and Thiébaux, 2001]. At the very least, these domains can support meaningful systematic experiments. Here we revisit one such artificial domain; the Blocks World. In visual perception, the basic physical and geometric constraints of our world play a crucial role. This idea goes back at least to Helmholtz and his argument for *unconscious inference*.

Larry Roberts argued that "the perception of solid objects is a process which can be based on the properties of three-dimensional transformations and the laws of nature" [Roberts, 1963]. Roberts' popular Blocks World was an early attempt to build a system for complete scene understanding for a closed artificial world of textureless polyhedral shapes by using a generic library of polyhedral block shapes. This toy domain has remained a staple of the AI literature for over 50 years.

TEOS is a Blocks World-based set of 3D object models with known complexity, controlled viewpoints, with a known level of self-occlusion. *TEOS* shares similarities in appearance with the so-called Shepard and Metzler objects [Shepard and Metzler, 1971], which are widely used in the literature for mental rotation tasks. See Figure 2.6 for an illustration of two such objects. Similarities are, for instance, the strict ninety-degree angle of elements making up an object, the use of only cuboids, the use of mainly one primitive (except for the base plate).

However, with *TEOS*, we present a set of objects that go beyond the Shepard and Metzler objects and aim to push the boundaries for computational as well as human experiments. Specifically, our objects have known, incrementally increasing complexity, they are designed to require that self-occlusion be solved, they share a common coordinate system, and we will show that they are

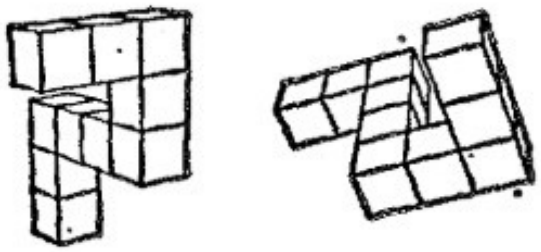


Figure 2.6: Example of the objects by [Shepard and Metzler, 1971] which are used as an inspiration to create the *TEOS* objects and as we advance, referred to as the Shepard and Metzler Objects. Displayed are two-dimensional projections of three-dimensional objects. The objects are assembled from a set of cubes, and each cube has a maximum of two neighbours, hence not allowing for branches. Source: [Shepard and Metzler, 1971]

challenging for visual tasks using modern classification algorithms.

In this section, we provide a brief review of existing object sets that allow for active observation, extend this to data sets that address occlusion, and present a number of approaches that deal with occlusion.

2.2.1.1 Sets of Objects for Three-Dimensional Observation

Scientifically established object sets that allow for three-dimensional observation, can be found mainly in biology [Martinho and Kacelnik, 2016, Srinath et al., 2021], psychology [Gauthier and Tarr, 1997, Burgundand and Marsolek, 2000], and in computer science [Johnson et al., 2017, Hinterstoisser et al., 2013, Hodaň et al., 2017, Lai et al., 2014]. However, they are scarce. In this work, we focus on objects that are not easily discriminable, which narrows the list even further down. For instance, in the field of computer vision, three-dimensional object datasets are often made up of "everyday objects", such as cup, box, duck, cat, phone [Hinterstoisser et al., 2013], which are usually easily distinguishable.

However, [Hodaň et al., 2017] proposed the *T-LESS* set of objects which cover thirty industry-relevant objects with no texture and no discriminative colour or reflectance. These characteristics make them much harder to tell apart, especially for a computational system. Figure 2.7 illustrates twelve images from this set of objects. Provided are a large number of images to train and test systems. The images are recorded with an RGB-D sensor and annotated with ground truth 6D

pose making *T-LESS* a testbed for pose estimation systems. Other testbeds for three-dimensional pose estimation include [Hinterstoisser et al., 2013, Lai et al., 2014].

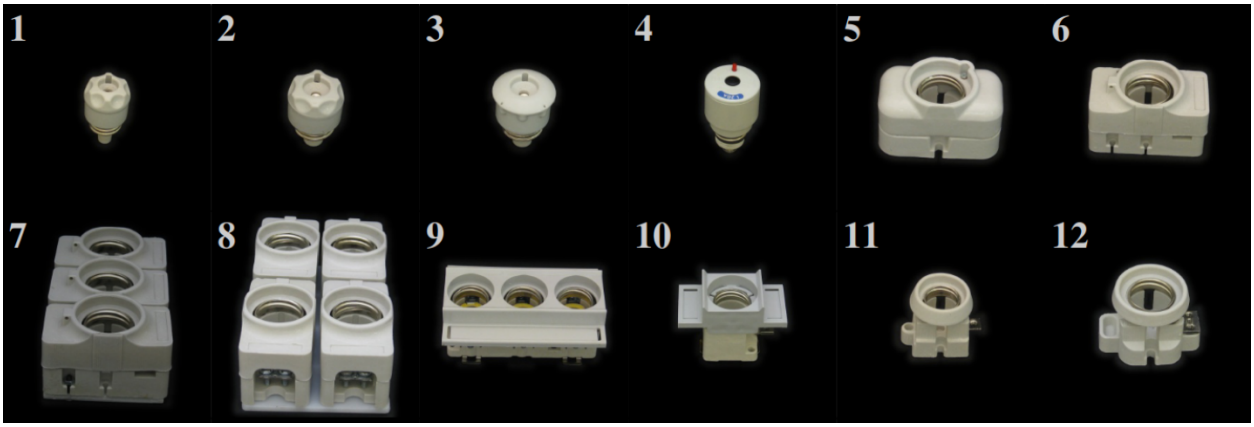


Figure 2.7: Examples of the *T-LESS* object set by [Hodaň et al., 2017]. Shown are the first twelve objects. Source: [Hodaň et al., 2017].

Separate from three-dimensional pose estimation, visual question answering (VQA) presents an image, usually depicting a scene of different objects, and asks natural language questions, such as "How many objects are either small cylinders or metal things?". Figure 2.8 shows an example of such a VQA scene taken from the *CLEVR* data set by [Johnson et al., 2017]. While extensions are proposed to change the viewpoint of the scene, this dataset falls into the realm of two-dimensional projections of three-dimensional scenes; hence, it does not allow for natural three-dimensional observation.

[Yamane et al., 2008] analyzed the neural activation for three-dimensional object shape in macaque brains. The goal of this study was to "disambiguate which three-dimensional shape factors are uniquely and consistently associated with neural responses." For this, random three-dimensional shape stimuli were constructed by extensively deforming a closed ellipsoidal surface (see Figure 2.9). These stimuli were rendered in depth by a combination of binocular disparity and shading cues and presented using stereoscopic depth.

[Gauthier and Tarr, 1997] presents the "Greebles Families". Shown in Figure 2.10 is a sample of a set of 60 control stimuli for faces. Each object belongs to a greeble family (shown are smar, osmit, galli, radok, tasio), gender (shown are plot and glip), and individual levels. The objects are used to explore mechanisms for face recognition. All objects are made up of four protruding parts

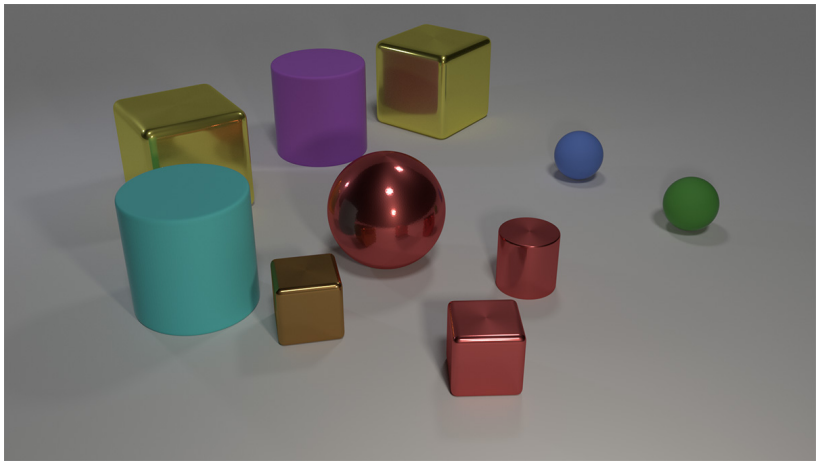


Figure 2.8: Examples of the *CLEVR* data set by [Johnson et al., 2017]. Shown is an example scene with different geometric shapes of different colors and material attributes. *CLEVR* is used to benchmark visual reasoning questions, such as "How many objects are either small cylinders or metal things?" Source: [Johnson et al., 2017]

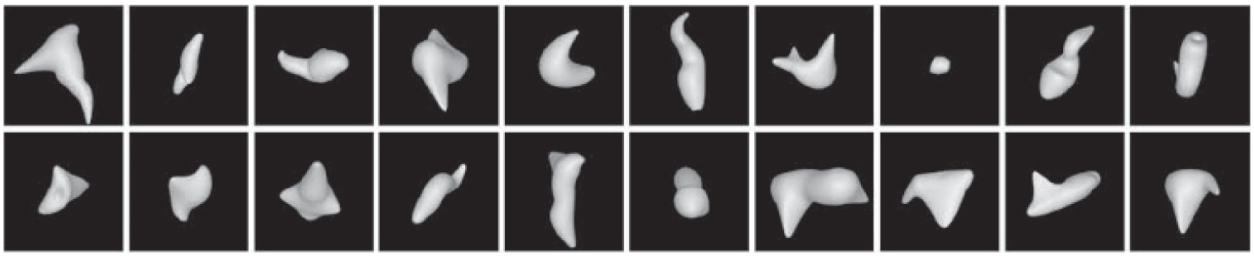


Figure 2.9: An excerpt of random three-dimensional shape stimuli that were constructed by extensively deforming a closed ellipsoidal surface. These stimuli were rendered in depth by a combination of binocular disparity and shading cues and presented using stereoscopic depth to trained rhesus monkeys. Source: [Yamane et al., 2008]

assembled in the same spatial configuration on a vertically oriented central path. Other "novel object" sets are [Barry et al., 2014, Sigurdardottir et al., 2018] which introduce the Fribbles and Yufos, respectively, to study psychology mechanisms including and beyond face recognition.

[Solbach et al., 2018] provide a polyhedral scene generator with controllable camera parameters and two different light settings. It is designed to enable research on how a program could parse a scene if it had multiple and definable viewpoints to consider. An example of a polyhedral scene from [Solbach et al., 2018] is shown from three different viewpoints in Figure 2.11 (top to bottom) with increasing complexity levels (left to right). The polyhedral scenes show a kind of "extreme" blocks world setting, feature significant self-occlusion. However, the space of possible objects and

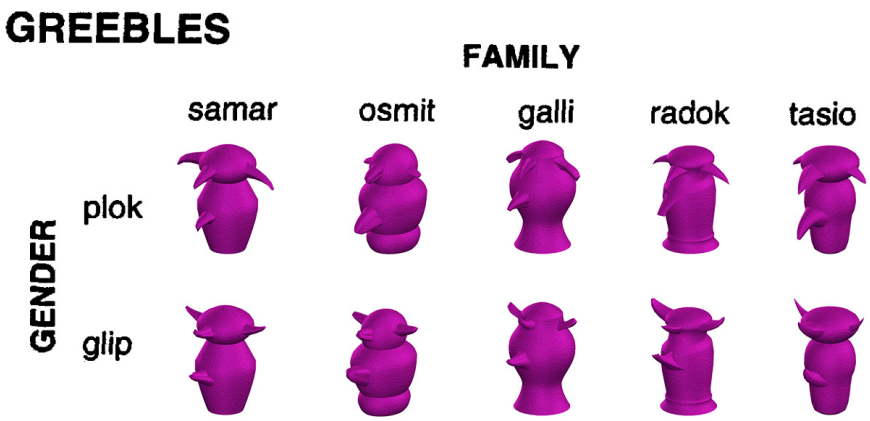


Figure 2.10: [Gauthier and Tarr, 1997] presents the "Greebles Families". Shown is a sample of a set of 60 control stimuli for faces. Each object belongs to a greeble family (shown are smar, osmit, galli, radok, tasio), gender (shown are plot and glip), and individual levels. Adapted from [Gauthier and Tarr, 1997] and using the Greebles Generator.

their characteristics are far too large to conveniently use in a learning scenario.

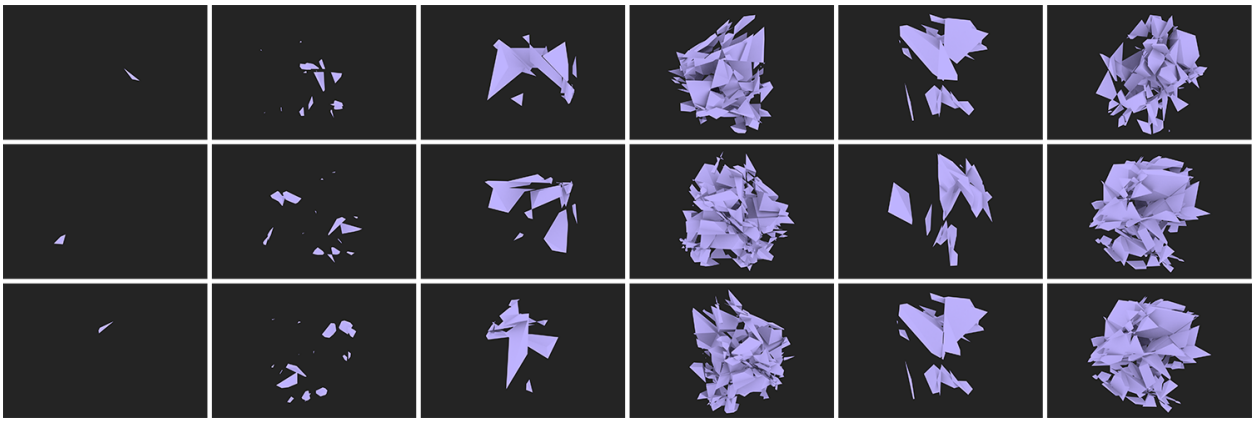


Figure 2.11: Six polyhedral scenes with increasing complexity (left to right) from three different viewpoints (top to bottom). The generator creates random scenes with known complexity characteristics and with verifiable properties. Source: [Solbach et al., 2018].

Lastly, in this brief overview, we point out an experiment in the field of evolutionary cognition. Here, a version of the same-different task is tested on trained ducklings. It is shown that they are able to imprint the relational concept of same-different. The experimental object pairs are geometric shapes (cube, sphere, cone, cylinder, and prism) and a variety of different colours (brown, red, green, purple, and off-white). Two different experiments were conducted, one for shape sameness and another for colour sameness. Figure 2.12 shows different experimental setups of the duckling

experiment – Middle: Example of the same colour training stimulus pair (blurry in the background). Right: A duckling correctly approaches and closely follows the novel same colour stimulus after being trained on it. These objects are obviously too simple for our scenario (minimal self-occlusion); however, the free-viewing ability of the experimental set up is desirable.

Figure 2.12: Example of the same-different experiments with ducklings. Portrayed are different experimental setups. Left: Example of a different shape stimulus pair. Middle: Example of the same colour training stimulus pair (blurry in the background). Right: A duckling correctly approaches and closely follows the novel same colour stimulus after being trained on it. Source: [Martinho and Kacelnik, 2016]

2.2.1.2 Occlusion Datasets

An intrinsicly complicating object characteristic is self-occlusion, however, it has not attracted much attention in the literature. However, occlusion caused by other objects has. A burden of deep learning is its need for vast amounts of training data. Even though occlusion and its effect on vision tasks have been addressed for some time, [Hsiao et al., 2010, Ouyang and Wang, 2012, Brachmann et al., 2014, Hsiao and Hebert, 2014], occlusion datasets created are usually too small to be used to train successful deep learning models. Furthermore, to our knowledge, datasets, if considering occlusion, mostly introduce various levels of clutter but fail to define occlusion in a generic way. For instance, the CMU Kitchen Occlusion dataset (CMU_KO8) by [Hsiao and Hebert, 2014] consists of 1,600 images of eight kitchen objects, which only yields 200 examples per class. The dataset has explicitly been designed to challenge object recognition algorithms with strong viewpoint and illumination changes, occlusions and clutter. Besides this, an occlusion reasoning module is also proposed (Section 2.2.1.3).

With the *ICCV 2015 Occluded Object Challenge* [Hinterstoisser et al., 2013, Brachmann et al., 2014], a dataset with eight objects positioned in a realistic setting of heavy occlusion is presented. The objects can be described as being of different domains (animals, office supplies, kitchenware, ...). However, neither a definition of occlusion nor a metric is given. Figure 2.13 shows an example image of the dataset.

Figure 2.13: A scene with different objects under occlusion from the *ICCV 2015 Occluded Object Challenge*.

The majority of occlusion datasets, however, deal with the occlusion of pedestrians. Specifically, in the context of autonomous driving, detecting pedestrians, even if occluded, is crucial to detect potential collisions. It is argued that most existing datasets are not designed for evaluating occlusion. For instance, the Caltech dataset [Dollár et al., 2012] only contains 105 out of 4250 images with occluded pedestrians. The CUHK Occlusion Dataset [Ouyang and Wang, 2012] is specifically designed as a pedestrian dataset with occlusion. The authors selected images from popular pedestrian datasets and recorded images from surveillance cameras and filtered them for occluded pedestrians. The dataset contains 1,063 images with binary classification to indicate occlusion. Other examples of occlusion datasets can be found in the person re-identification literature [Zheng et al., 2015, Miao et al., 2019]. The goal is to re-identify a target person after they have disappeared due to occlusion of, for instance, other people, objects or left the camera view.

2.2.1.3 Occlusion Reasoning

Reasoning about occlusion has been used in many areas, from object recognition to tracking and segmentation. Reported in [Hsiao and Hebert, 2014], the literature is extensive, but there has been comparatively little work on modelling occlusion from different viewpoints and using three-dimensional information until recently. Further, occlusion reasoning is broadly classified into five categories; inconsistent object statistics, multiple images, part-based models, three-dimensional reasoning, and convolutional neural networks.

The first category uses inconsistent object statistics to reason about potential occlusion. For instance, [Meger et al., 2011] use inconsistencies in three-dimensional sensor data to classify occlusions. [Girshick et al., 2011] introduce an occluder part in their grammar model when all parts cannot be placed. [Wang et al., 2009] use a scoring metric based on individual HOG filter cells. [Hsiao and Hebert, 2014] incorporate occlusion reasoning in object detection in a two-stage manner. First, in a bottom-up stage, occluded regions are hypothesized from image data. Second, a top-down stage is used that relies on prior knowledge to score the candidates' occlusion plausibility. Extensive evaluation on single and multiple views shows that incorporating occlusion reasoning yields improvement in recognizing texture-less objects under severe occlusions.

The use of multiple images characterizes the second category. For these approaches, consecutive images are necessary to disambiguate the object from occluders. For instance, [Ess et al., 2009] detects the objects and extrapolates the state of occluded objects using an Extended Kalman Filter. Reliable tracklets that are used in a temporal sliding window fashion are generated to disambiguate occluded objects in [Xing et al., 2009].

One of the largest categories is part-based model approaches. A challenge of global object templates is occlusion as their performance degrades with its presence significantly. A popular solution to this problem is to separate the object into a set of parts and detect parts individually. This approach yields more robust detections with respect to occlusion. For example, [Shu et al., 2012] analyzes the contribution of each part using a linear SVM and trains the classifier to use unoccluded parts to maximize the probability of detection. [Wu and Nevatia, 2009] go a step further and use multiple part detectors to maximize the joint likelihood. Binary classification of parts is introduced by [Vedaldi and Zisserman, 2009]. They decompose the HOG descriptor into

small blocks that selectively switch between an object and an occlusion descriptor.

More recent work uses three-dimensional information. [Pepikj et al., 2013] train multiple occlusion detectors on mined three-dimensional annotated urban street scenes that contain distinctive, reoccurring occlusion patterns. [Wang et al., 2013] use RGB-D information and an extended Hough voting to include object location and its visibility pattern. [Radwan et al., 2013] addresses precisely the problem of self-occlusion in the context of human pose estimation and adds an inference step to handle self-occlusion to an off-the-shelf body pose detector to increase its performance under self-occlusion. [Bonde et al., 2014] propose an object recognition system that also works in the presence of occlusion and clutter. They use a soft label *Random Forest* to learn the shape features of an object. Using occlusion information, taken from the depth data, the forest emphasizes the shape, thus making it robust to occlusion. More recently, [Sahin et al., 2019] proposes a part-based architecture to recover the 6D object pose in-depth images that is also able to deal with occlusion. Their *Intrinsic Structure Adaptor* adapts the distribution shifts arising from shape discrepancies and removes the variations of texture, illumination, pose, etc.

Convolutional neural networks form the last group of approaches. [Reddy et al., 2019] introduces a framework to predict two-dimensional and three-dimensional locations of occluded key points for objects to mitigate the effect of occlusion on the performance. Evaluated on CAD data and a large image set of vehicles at busy city intersections, the approach increases the localization accuracy of MaskRCNN by about 10%. A self-occlusion example can be seen in Figure 2.14. [Li et al., 2019] uses deep supervision to fine-grain image classification. In their approach, they simulate challenging occlusion configurations between objects to enable reliable data-driven occlusion reasoning. Occlusion is modelled by rendering multiple object configurations and extracting the visibility level of the object of interest. [Kortylewski et al., 2021] introduce CompositionalNets, which is combined with part-based models. The fully-connected classification layer is replaced with a differentiable compositional model. The idea is to decompose images into objects and context and then decompose objects into parts and objects' pose.

The approach can learn occlusion invariant features and discard occluders during classification, hence increasing performance under occlusion. However, a trade-off is that a good occluder localization lowers classification performance because classification benefits from features that are invariant to occlusion, where occluder localization requires a different type of features. Namely,

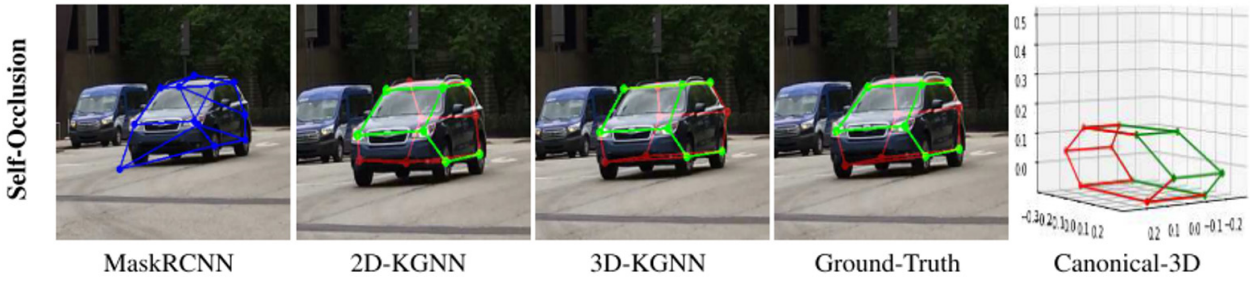


Figure 2.14: The effect of occlusion reasoning used in a CNN. Left the original CNN (MaskRCNN) and different (two-dimensional and three-dimensional) occlusion reasoning approaches improve the detection [Reddy et al., 2019].

ones that are sensitive to occlusion. It is pointed out that it is essential to resolve this trade-off with new types of models.

2.2.2 Object Definitions

None of the object sets presented in Section 2.2.1 fit all of our needs for a three-dimensional version of the same-different task that allows for active observation. Specifically, the set of objects needs to satisfy the following characteristics:

- Known complexity
- Common coordinate system
- Self-occlusion
- Real and three-dimensional
- No familiar objects, hence objects that naturally encourage active observation

With *TEOS*, we present in total 48 objects, split into two sets; L_1 and L_2. L_1 consists of 36 objects in 18 complexity classes, hence tailored towards research exploring the effect of finely grained complexity changes.

All objects consist of the following two elements: One 20mm x 60mm x 120mm base (Figure 2.15 right) and n 20mm x 20mm x 60mm cuboids (Figure 2.15 left).

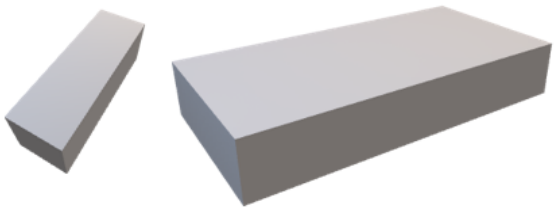


Figure 2.15: The building blocks used to create the objects of *TEOS*; cuboid (left) and base (right).

The complexity of an object is simply calculated as

$$compl = n + 1 \tag{2.1}$$

Where n is the number of cuboids used[4]. Further, *inter-class object complexity* refers to objects that are not of the same object complexity class, while *intra-class object complexity* refers to objects that are of the same object complexity class but differ in their configurations.

Building an object, the base has five connection points for cuboids. All cuboids are only attached upright, sitting flush with the bottom of the base. This also makes it simple to define a coordinate system.

All objects share the same coordinate system, which is crucial for any research that looks at the effect of the orientational difference of three-dimensional objects. The coordinate system is defined as depicted in Figure 2.16 (left); the Y-Axis is orthogonal to the base, the X-Axis is orthogonal to the Y-Axis and parallel to the base through its center of gravity, and the Z-Axis is orthogonal to both the Y- and X-Axis with the positive direction through the side of the base with two cuboid connections. Every object shares the characteristics of two cuboid connections on one side and three connections on the other on the base (Figure 2.16 (right)).

A base has five connectors at which a cuboid can be attached (Figure 2.16 (right)). Consecutive cuboids are always orthogonally and never aligned in their direction, which is one of the differences to the Sheppard and Metzler objects. Furthermore, cuboids never intersect or touch neighbouring cuboids, hence avoiding geometrical loops. Creating the objects for L_1, we focused on making the complexity comparable by consecutively adding one cuboid per complexity class to the object of

[4]One might think a more accurate measure would be to consider the size and number of faces. However, the upcoming rules of combinations preclude this.

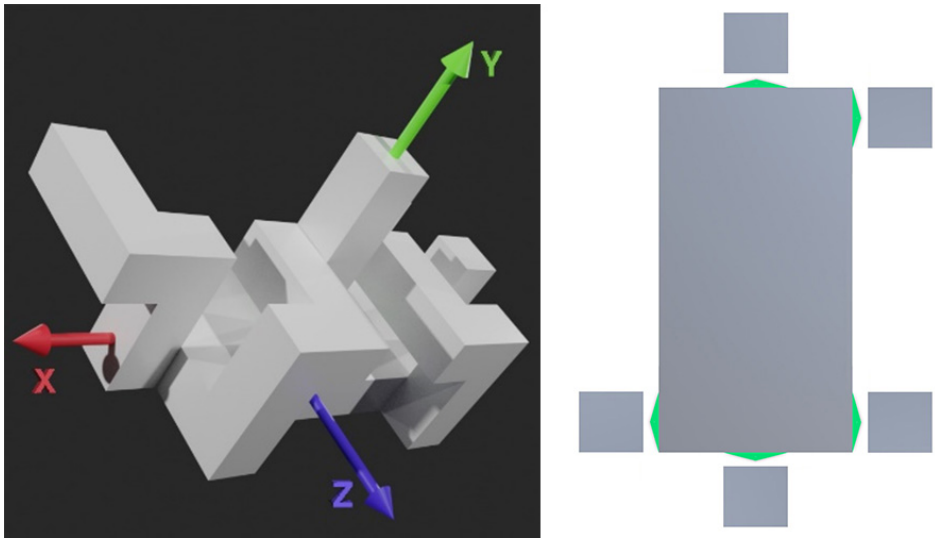


Figure 2.16: Left: Illustration of the common coordinate system of the objects. Right: Possible cuboid connection points on the base.

the previous complexity class.

In several empirical studies with human subjects, we have studied the relationship between the number of elements per object and classification accuracy. The performance to classify L_1 objects is reliable (accuracy of $> 98\%$) for objects of $compl = 7$ (Eq. 2.1). The classification is less accurate (89%) with objects of $compl = 10$. Finally, the classification gets challenging (57%) with objects of $compl = 18$.

Based on these findings, we have created the L_2 set. It is designed with less variation across complexity classes but more variation within a complexity class. Twelve objects are evenly split into three complexity classes; easy with seven elements, medium with ten elements, and hard with 18 elements. Within a complexity class, the objects only differ in one small detail by changing one of the elements' orientation. This said, the L_1 and L_2 subsets enable two classes of self-occlusion analysis; one with high inter-class object complexity variability and another with high intra-class object complexity variability in appearance, respectively. Furthermore, as will be discussed later, this set also provides self-occlusion distributions with unique means for each of the three complexity classes (see Figure 2.23).

The L_1 objects can be seen in Figure 2.17 (top), and consist of 36 objects split into 18 complexity classes. There is a distractor object of the same complexity for each object that differs only in one

small detail; one of the items is oriented differently. The introduction of the distractor objects is intended to support research in visual recognition, where merely counting the number of elements would reveal the object class. The L_2 objects can be seen in Figure 2.17 (bottom).

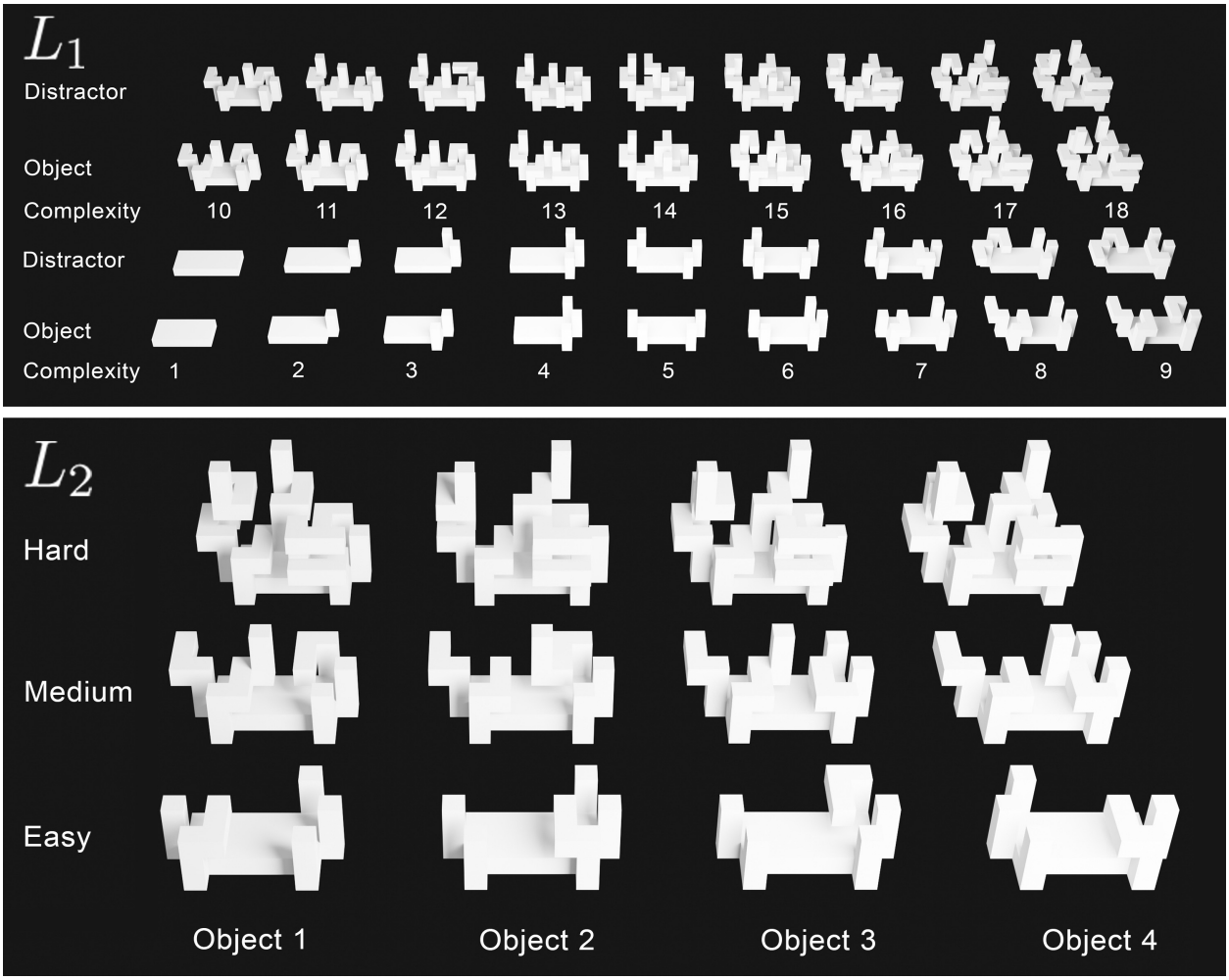


Figure 2.17: Top: Illustration of L_1 with all 36 objects. Bottom: Illustration of L_2 with all 12 objects, split into three different complexity classes.

2.2.3 Dataset Acquisition

TEOS is a dataset that is designed to be used in the virtual as well as the real world. For the former, one can use the rendered images and provided three-dimensional Models (.STL file). For the latter, the objects are designed to be printable with a 3D printer. However, in this section, we want to focus on the generation of the rendered dataset images for which we have used Blender

[Community, 2018], a free and open-source three-dimensional computer graphics software toolset. For *TEOS*, each object was rendered from 768 views, totalling 36,864 images. To achieve realistic renderings of the objects, we used the Cycles Path Tracing rendering engine, created a white, smooth, plastic imitating material, set six light sources in the rendering scene and used 4,096 paths to trace each pixel.

Each object is rendered from the same set of views. To determine the views, we used the Fibonacci lattice [Stanley, 1988] approach. This approach allows distributing points on a sphere approximately evenly. Other techniques, for example, using radial distance, polar angle and azimuthal angle, will result in an unevenly sampled sphere; dense on the poles and sparse closer to the equator. Figure 2.18 illustrates the chosen views to generate the dataset. Each blue-coloured point represents a location where the camera is placed and oriented to the center where the object (red) is. We chose a sphere radius of two such that the object is view-filling but not cropped.

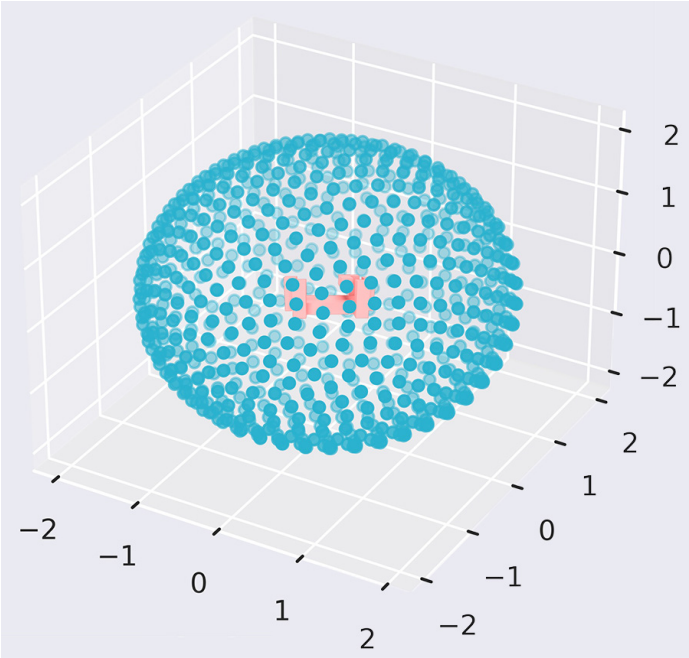


Figure 2.18: Illustration of viewpoints used to render each object of L_1 and L_2. Views are evenly distributed on a sphere around an object (blue points) and point towards the object (light red). In total 768 views are taken.

Further, as it is sometimes practiced in the machine learning community [Everingham et al., 2010, Lin et al., 2014, Matthey et al., 2017, Kabra et al., 2019], we also provide the object mask and renderings with a dark and bright background for data augmentation purposes. The annotation

file contains the object-type, view-id, bounding box information, object and camera positions and orientations, object dimensions, and self-occlusion value.

2.2.4 Self-Occlusion Measure

It seems evident that if we see less of an object, it is harder to classify it. Regions of the object that are occluded to a viewer might hold distinct features to tell object X apart from object Y. In other words, occlusion for visual classification plays an important role. However, it is not only dependent on the view but also on the object. Let us take, for example, a sphere. No matter from which angle we look at it, we always observe 50% of it. On the contrary, for a complex polygonal shape, this cannot be answered as quickly as it is dependent on its geometry.

[Gay-Bellile et al., 2010] distinguishes two kinds of occlusions; "external occlusion" and "self-occlusion." "External occlusion" is caused by an object entering the space between the camera and the object of interest and "self-occlusion" which describes the occlusion caused by the object of interest to itself. For *TEOS*, we are interested in the latter, as we always have one object in the scene.

To our knowledge, no standard self-occlusion measure is used for computational approaches; therefore, we aim to specify our own intuitive measure as:

$$SO_{c_i} = \frac{A_\phi^{c_i}}{A_\sigma} \tag{2.2}$$

Where A_ϕ is defined as the occluded (not visible) surface area of the object, A_σ stands for the total surface area of the object, and c_i for the camera pose. In our rendered dataset, the self-occlusion was calculated by using the following algorithm, but note that for this calculation, the object identity must be known:

Algorithm 1 Self-Occlusion

1: Iterates over all faces of the object with valid normals and calculate the (A_σ)
2: Subdivide the objects into a few thousand elements
3: Position the camera at a given location and pointing it at the object (see Figure 2.18)
4: Select vertices that are visible through view-port
5: Divide object into visible and not-visible part
6: Iterate over all faces of the not-visible object with valid normals and calculate (A_ϕ)
7: Lastly, calculate Self-Occlusion (Equation 2.2)

Applying this to the L_1 set of objects. We can see in Figure 2.19 that an increase of complexity also increases the average amount of self-occlusion among all viewpoints.

Each point shows the self-occlusion of the respective object from a specific viewpoint. The viewpoints are evenly distributed on a sphere around an object, resulting in 768 unique views. The straight line illustrates the increase in average self-occlusion as the complexity increases. However, worth noting, with an increasing amount of complexity, the self-occlusion distribution per class decreases.

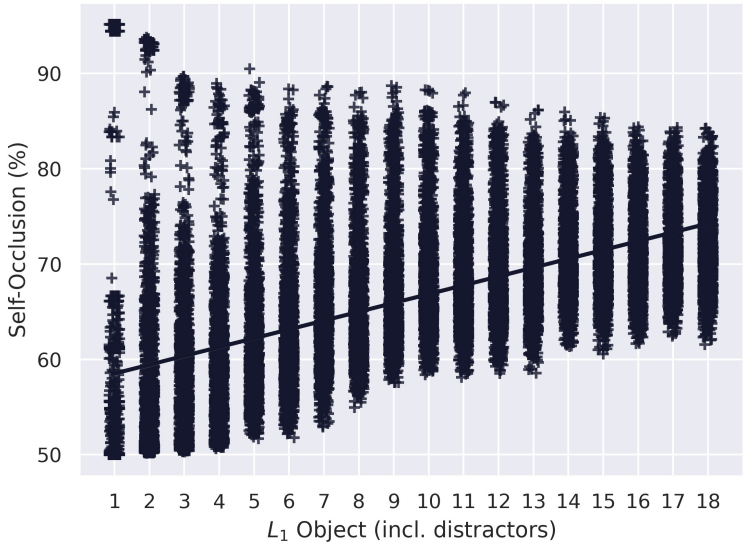


Figure 2.19: Illustration of the amount of average self-occlusion per object of L_1. Each point shows the self-occlusion of the respective object from one of the 768 viewpoints. The straight line illustrates the increase in average self-occlusion as the complexity increases.

An object might have different views from which it causes the same amount of self-occlusion, resulting in perhaps a considerably different appearance. Figure 2.20 shows an example of two objects from two different views with the same amount of occlusion.

Therefore, we also consider the camera's point of view with c_i as the camera pose. Here, c_i is defined as the camera position $c_i = (x_i, y_i, z_i)$ and computed based on the Fibonacci lattice approach (see Figure 2.18). The camera orientation is automatically set such that the object is in the centre of the viewpoint.

For evaluation purposes, we also define a function that maps a camera position (c_i) onto one of the eight regions of the octahedral viewing sphere placed at the centre of an object. Figure 2.21

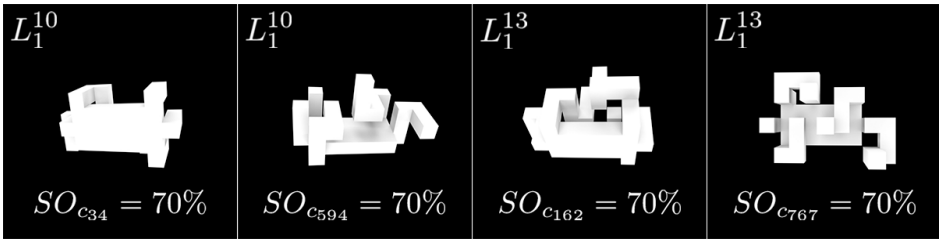


Figure 2.20: Examples of different objects (Object 10 and 13 of L_1) and poses causing the same amount of occlusion but different appearances.

illustrates a mapping example for two camera-positions.

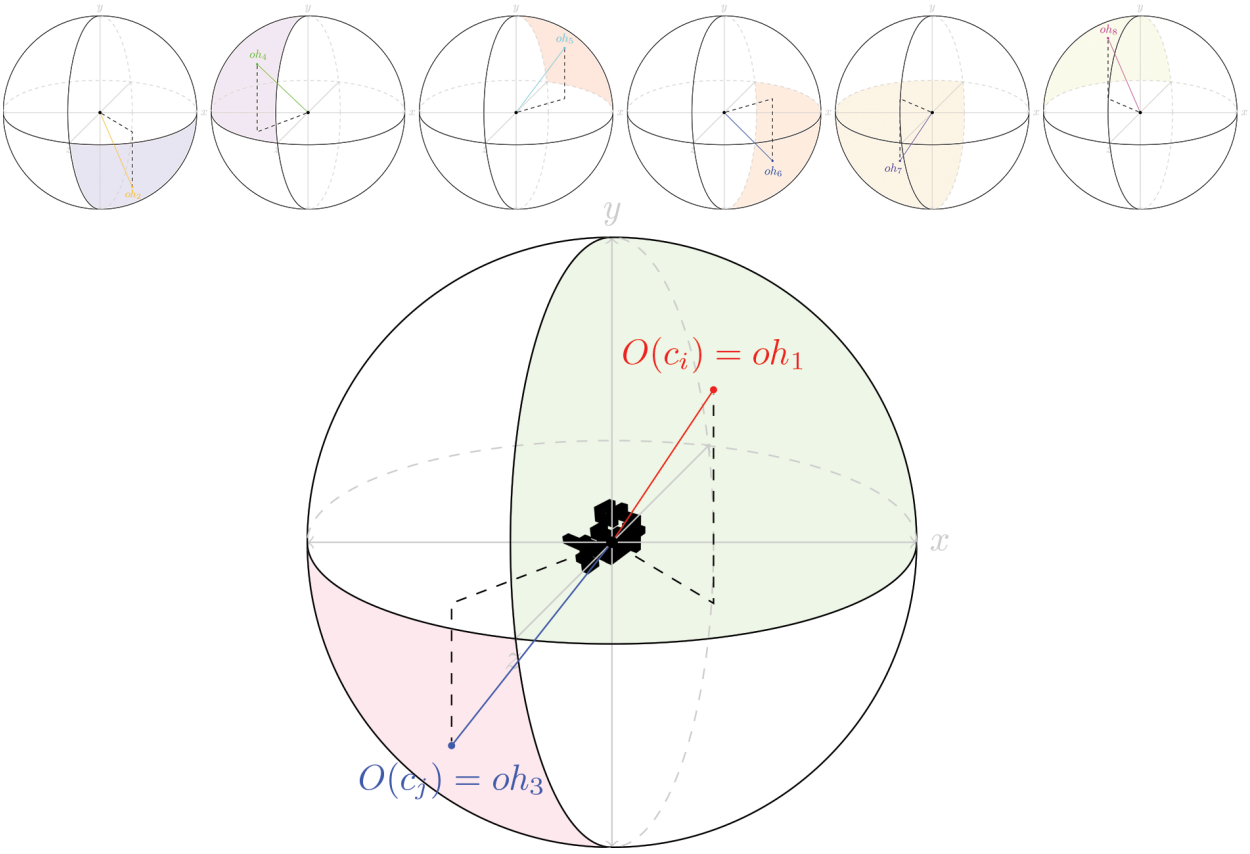


Figure 2.21: Visualization of the octahedral sphere based projection used to map camera positions. Bottom: two example camera poses (c_i and c_j) mapped to oh_1 and oh_3.

We represented the viewing sphere around an object as a spherically tiled octahedron, resulting in eight uniformly distributed triangles. To map a viewpoint c_i to a tile, we perform a determinant check to see in which tile a given camera pose c_i is located.

Figure 2.22 shows eight examples of the same object (object-7) from different viewing angles

and sorted based on their amount of self-occlusion. As can be seen in the illustration, a single object can cast many different appearances based on the viewing angle and a significant change in the amount of what is observable of it.

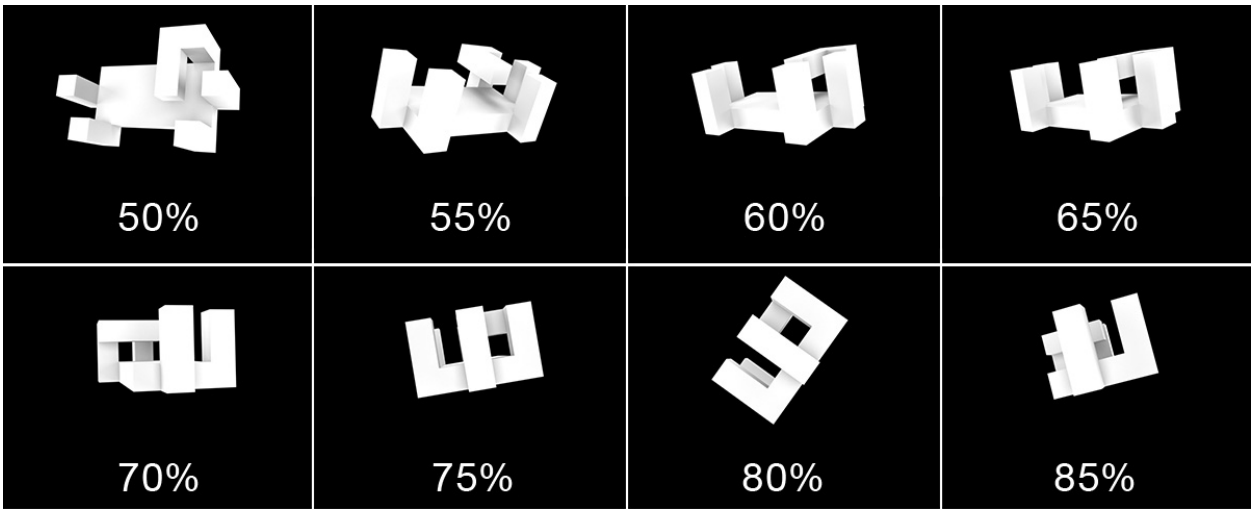


Figure 2.22: Some object viewpoints and their corresponding SO_{c_i}.

Figure 2.23 illustrates the self-occlusion distribution for L_1 and L_2 (top) and the distributional relation between viewpoint mapping and self-occlusion for L_1 and L_2 (bottom). Self-Occlusion for L_1 ranges from 49.99% to 95.16% with a mean at around 68% and L_2 from 54.08% to 87.5% with a mean at 61% (Easy), 63% (Medium), and 71% (Hard). The lower half of the figure shows that different octahedron viewpoints result in varying amounts of self-occlusion. For both L_1 and L_2, an overall sweet-spot with the least self-occlusion is at oh_5, presumably resulting in the best classification result. More specifically, for L_2 class "Hard", this spot is at $oh_{2/4}$ and for class "Medium" at oh_7.

2.2.5 Baseline Evaluation

In this section, we discuss how well modern classification approaches perform on *TEOS*. We have chosen five deep learning models with different properties, carefully trained and evaluated them on *TEOS*.

We have chosen Inception-V3 [Szegedy et al., 2016], MobileNet-V2 [Sandler et al., 2018], ResNet-V2 [He et al., 2016], VGG16 [Simonyan and Zisserman, 2014] and EfficientNet [Tan and Le,

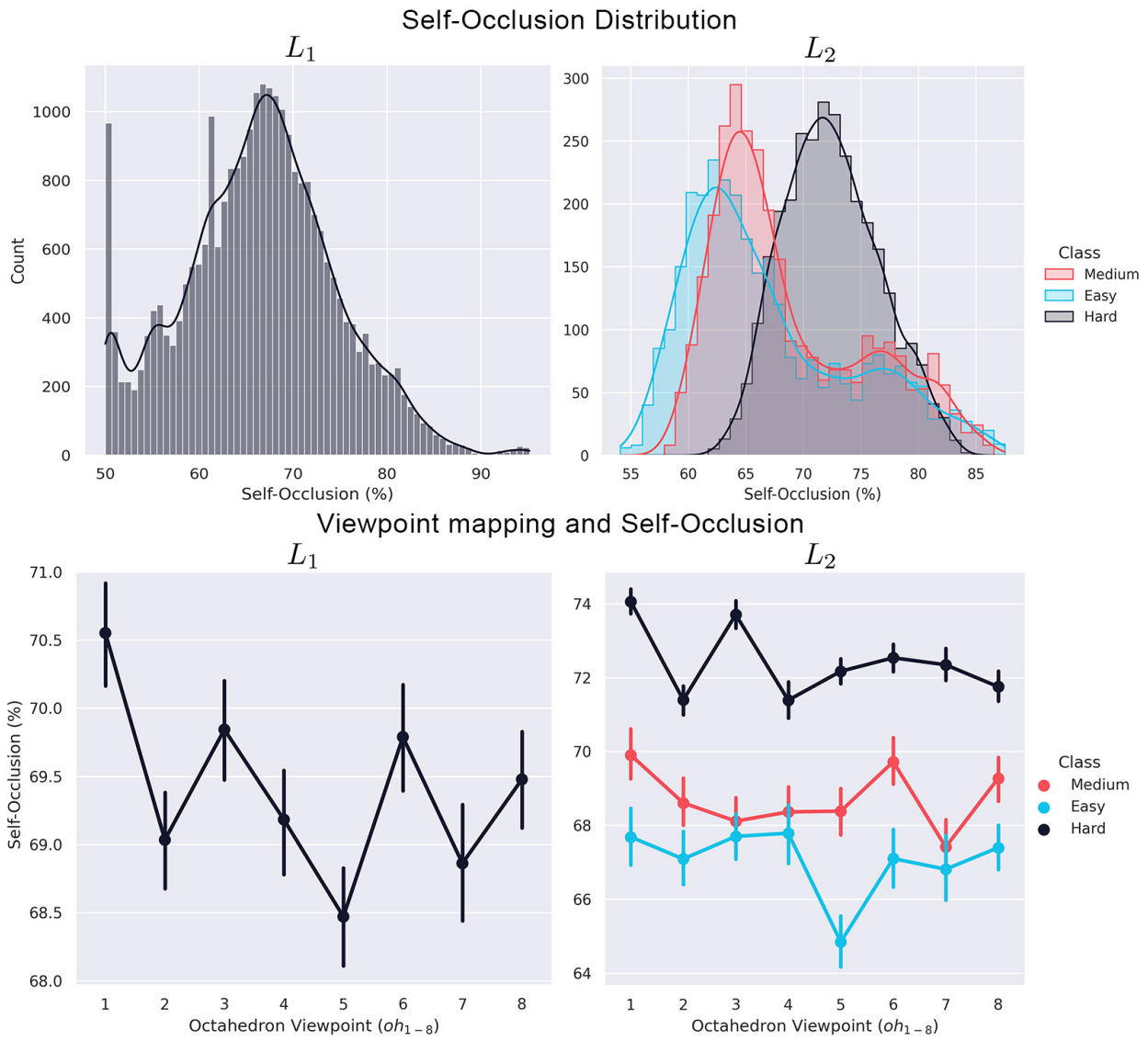


Figure 2.23: Illustration of the self-occlusion distribution for L_1 and L_2 (top), as well as the distributional relation between viewpoint mapping and self-occlusion for L_1 and L_2 (bottom).

2019] as reference networks for *TEOS*. Their trained version of *TEOS* is made publicly available: `https://data.nvision.eecs.yorku.ca/TEOS`. Table 2.1 shows more details about the networks in ascending order of their parameter count.

Besides the architecture of CNNs, a crucial element is the choice of training parameters and so-called hyperparameters. In our case, we have looked at the input size, input noise, dropout rate, learning rate, optimization algorithm and lastly, the difference between learning from scratch and fine-tuning the networks. Hyperparameters such as input noise, drop rate, learning rate were

Table 2.1: High-Level CNN Characteristics

CNN	Layers	Parameters (mil.)
MobileNet-V2	53	3.4
Inception-V3	48	24
ResNet-V2	152	58.4
EfficientNet-B7	813	66
VGG16	152	138

determined using the hyperparameter optimizer Hyperband by [Li et al., 2017]. All networks are pretrained on ImageNet [Deng et al., 2009]. This might have an effect on the performance as the images are distinctly different compared to the ones from *TEOS*. The remaining parameters were empirically determined. Table 2.2 presents the parameters used to establish the baseline of *TEOS*.

Table 2.2: Training Parameters

Parameter	Value
Input Size	224 x 224 – 800 x 800 (dependent on CNN)
Input Noise	Gaussian Noise of 0.1
Drop Rate	20%
Learning Rate	1e-5
Optimizer	Adam Optimization [Kingma and Ba, 2015]
Pretrained	ImageNet [Deng et al., 2009]
Learning Method	Fine Tuning

To prepare the data for training, we chose a 15/15/70 split. Where 15% of the dataset was allocated for validation, 15% for testing, and augmented the remaining 70% with the following data augmentation techniques [Shorten and Khoshgoftaar, 2019]: rotation ($0 - 40°$), width/height shift (0-20%) and zoom (0-20%).

Our results show that MobileNet-V2 performed best across L_1 and L_2. Specifically, for L_1, it achieved a top-1 accuracy of 17.25% and 10.83% on the L_2 data set. See Figure 2.24 for the classification accuracies of all networks. It seems that MobileNet-V2 is the only network that was able to learn some aspects of *TEOS*, performing with a large (L_1) or small (L_2) margin above chance, whereas all other networks perform at around chance. This, perhaps, has something to do with the relatively homogeneous appearance of *TEOS*, not allowing the more complex CNNs to learn from. However, this needs to be investigated further in the future.

Generally, L_2 is more challenging to learn for CNNs than L_1. Even the best performing CNN

is only 2.53% above chance, where this margin for L_1 was at about 14.5%. This is explainable with the high intra-class similarity of L_2 – objects of one class look very similar to each other and only vary in a small detail, which might be only observable from certain views, hence will be confused with each other.

The L_1 dataset, on the other side, has a low inter-class similarity – the appearance of objects varies between classes. A closer look at the results of L_2 reveals that more extensive networks (VGG16 and EfficientNet-B7) were able to learn objects of class "Hard" of L_2; however, they could not learn "Medium" and "Easy" Objects. The smaller networks, on the other hand (MobileNet-V2 and Inception-V3), were able to learn "Easy" and "Medium" objects but not "Hard." Except for MobileNet-V2, all networks have problems learning the "Easy" Objects. See Figure 2.24 for details.

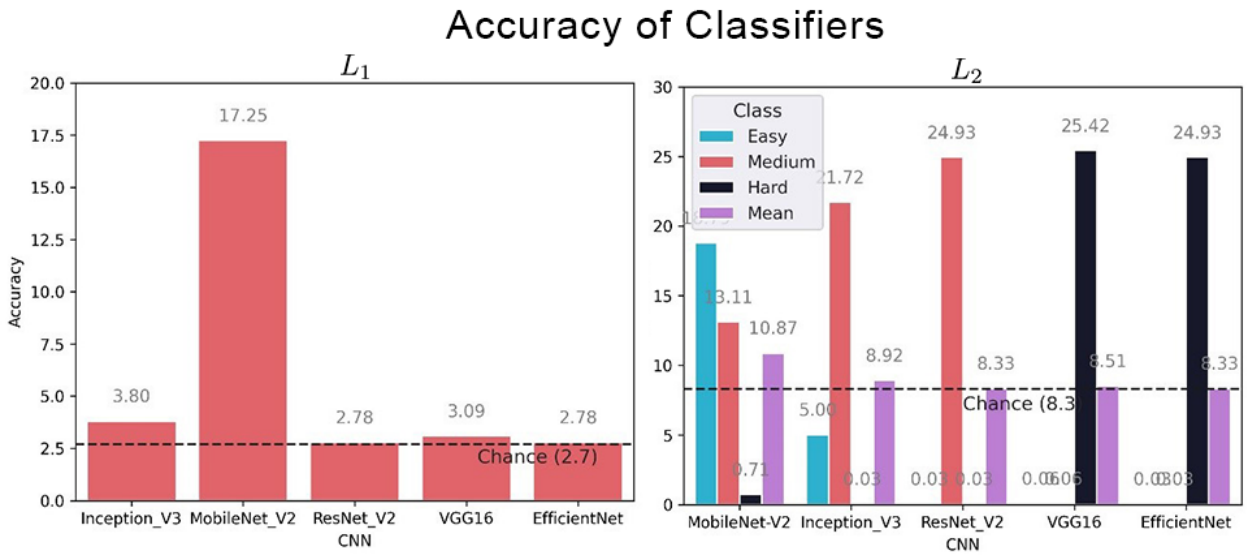


Figure 2.24: Evaluation results on L_1 (left) and L_2 (right) for five different CNNs and their accuracy across the entire datasets.

Regarding the connection between classification accuracy and the amount of self-occlusion, it can be generally said that the classification accuracy goes down if self-occlusion increases. We have chosen the three best-performing CNNs to analyze this connection and grouped L_1 and L_2 from 50% to 85% self-occlusion in 5% intervals. $< 50\%$ captures viewpoints with a self-occlusion of less than 50%. $> 85\%$ includes images with more than 85% (Figure 2.25).

Furthermore, we also investigated the connection between the viewpoint mapped to an octahedral viewing-sphere and accuracy. As can be seen in the example of L_1 and MobileNet-V2,

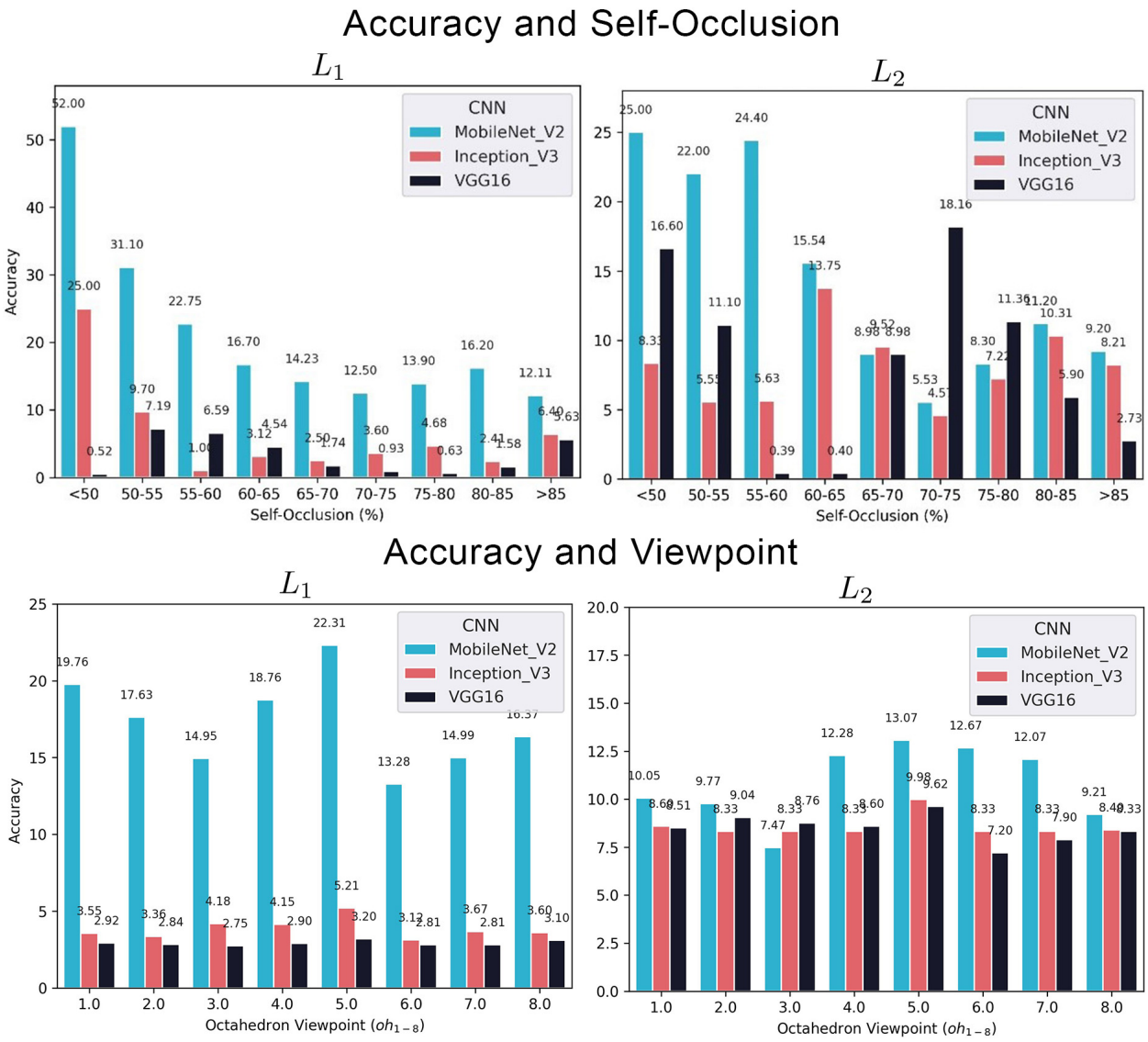


Figure 2.25: Evaluation for the three top-performing CNNs. Top: Accuracy across the entire datasets with respect to self-occlusion. Bottom: Accuracy and how it is affected by the chosen viewpoint.

the viewpoint does play a vital role and can result in an increase of accuracy performance by $13.28 \hookrightarrow 22.31 = 67.99\%$. Across L_1 and L_2 the octahedral viewpoint resulting in the best performance was oh_5. This can be explained with that all objects share a common coordinate system and shows once more that the viewpoint matters and, even more, that an *ideal viewpoint* can exist.

Further, even though the CNNs are trained and validated on the entire data set, their best performance can be seen at lower self-occlusion rates, which shows the vital role of self-occlusion

for object classification performance.

2.2.6 Conclusion and Future Directions

In this section, we have presented a novel three-dimensional blocks world dataset that focuses on the geometric shape of three-dimensional objects and their omnipresent challenge of self-occlusion. We have created two data sets, L_1 and L_2, including hundreds of high-resolution, realistic renderings from known camera angles. Each data set also comes with rich annotations.

Further, we have presented a simple but precise measure of self-occlusion and were able to show how self-occlusion challenges the classification accuracy of modern CNNs and the viewpoint can benefit the classification. Lastly, in our baseline evaluation, we have presented that CNNs cannot learn *TEOS*, leaving room for future work improvements.

In an additional study, we show that active control over the input is crucial to increase classification results [Korbach et al., 2021]. There, we use deep reinforcement learning to control the next-best view and show that the classification accuracy can be increased to 96.33% with an average of 4.28 additional views. However, only a selection of the L_1 objects (no distractors) and a simplified appearance (no shadowing) was used.

With this set of objects, we have created challenging stimuli with different levels of self-occlusion for the three-dimensional same-different task. Furthermore, we hope that *TEOS* is useful for research in the realm of active vision – to plan and reason for the next-best-view seems to be crucial to increase object classification performance; this has been partially already shown in [Korbach et al., 2021].

In the next section with present a first of its kind experimental set up that we have designed and built to precicely track human subjects performing tasks using the *TEOS* object sets.

2.3 *PESAO* – Psychophysical Experimental Set Up for Active Observers

We presents a novel experimental set up for active, visual observers. It is the first of its kind and allows for precise head- and gaze-tracking of human subjects while completely untethered, only wearing a pair of glasses and a small processing unit. Figure 2.26 shows a subject using

the experimental set up. This set up will be used to conduct a large-scale study (Chapter 3) to investigate human visual behaviours for the three-dimensional same-different task.

Figure 2.26: A close-up of a subject using *PESAO*. The subject approached an object while wearing the eye-tracking glasses with the motion tracking mount.

The section is divided into a brief background about existing systems that track gaze and head motion (Section 2.3.1), an overview about our system dubbed **P**sychophysical **E**xperimental **S**et Up for **A**ctive **O**bserver or short *PESAO*. In Section 2.3.2, full detail on the hardware used to build our system (Section 2.3.3), how we set up the hardware (Section 2.3.4), the software suite that runs experiments, collects and processes data and more, called *PESAOlib*, in Section 2.3.5. Lastly, a summary is provided in Section 2.3.6.

2.3.1 Background

Most past and present research in computer vision involves passively observed data. Humans, however, are active observers outside the lab; they explore, search, select what and how to look [Bajcsy, 1988]. Nonetheless, how exactly active observation occurs in humans so that it can inform the design of active computer vision systems is an open problem.

Here, we focus on the active and externally observable part of the process – head and gaze. To understand how active visual observation in humans occurs, one needs to monitor the actions performed to solve the visual task. This can be done in different ways and usually depends on the purpose of the system. At one end of the spectrum of tracking-possibility is simply observing the actions and taking notes with a pen and paper. The other end of the spectrum is using high-resolution sensors that operate at microsecond speeds that provide precise tracking information. With *PESAO* we aim for the latter.

Other systems to track active observers have been proposed – all with their own particular goals. For instance, [Khamis et al., 2017] presents an active eye tracker, *EyeScout*, for large interactive public displays. The focus of the system is to track random people without any interaction. Figure 2.27 shows a sketch of the system. To realize EyeScout, an eye-tracking device is mounted on a rail system which moves according to the body tracking information collected by the body tracker from across the room. The limitation of the system is that it only moves along a linear rail making it impossible to track the gaze if the person turns away from the eye tracker. Further, the tracking accuracy depends on the distance and height of the user. However, future versions are said to be able to deal with this by adjusting the eye tracker's angle dynamically.

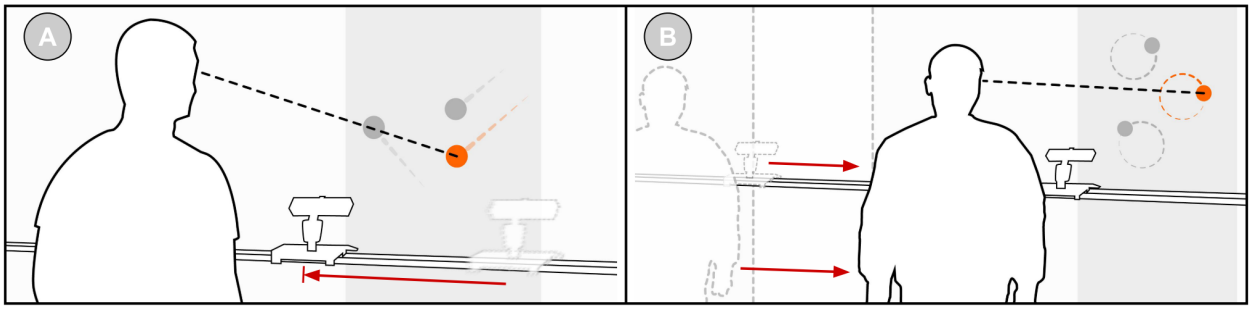


Figure 2.27: EyeScout is an active eye-tracking system that enables gaze interaction with large public displays. It supports two interaction modes: In "Walk then Interact," the user can walk to a location in front of the display and the system positions itself accurately to enable gaze interaction (A). In "Walk and Interact" the user can walk along with the display, and the system follows the user, thereby enabling gaze interaction while on the move (B). Source: [Khamis et al., 2017]

With *Gaze-in-wild* [Kothari et al., 2020] the authors propose a head and eye-tracking system that allows for full 6 degrees of freedom tracking. It combines eye-tracking glasses, first-person camera, IMU and a stereo camera to track eye and head movements seemingly. The system is used to collect a dataset on various tasks, such as indoor navigation, ball catching, visual search, tea

making, and more (see Figure 2.28).

Figure 2.28: Task selections in the GW dataset. Left to right: Indoor navigation, ball catching, visual search and tea making. Source: [Kothari et al., 2020]. Consent provided as Open Access: Creative Commons Attribution 4.0 International. To view a copy of this license, visit `https://creativecommons.org/licenses/by/4.0/`.

The participant wears in total three items; eye-tracking glasses with first-person camera, a hardhat with a mounted stereo camera and IMU, and a backpack with a laptop. This set up is quite similar to the first prototype of *PESAO* (Figure 2.29).

Figure 2.29: First prototype of *PESAO* (July, 2018).

However, the limitations listed by [Kothari et al., 2020] are in accordance with one of our prototype, and helped us step away from a solution considering a hardhat and laptop backpack for *PESAO*. The eye-tracker, IMU and stereo camera need to be calibrated to align the data. The movement of the subject can result in slippage of the hardhead or even the eye tracker causing misalignment of the tracking streams. Another issue is the inside-out[5] tracking using the stereo camera. With one of our first prototypes, we have explored inside-out tracking as well (Figure 2.29) and found that, not surprisingly, inside-out tracking using a stereo camera requires an environment with rich visual features to work. However, the *PESAO* environment is purposefully built with few visual features other than the actual stimuli. Rendering is a challenging environment for inside-out tracking using a stereo camera due to inaccurate tracking (patchy and lost tracking and drift).

Furthermore, our experimental set up the environment does not provide rich visual features to allow for three-dimensional reconstruction of the stereo data, causing incomplete and inaccurate positional tracking. Other approaches that use inside-out tracking are [DuTell et al., 2021, Hausamann et al., 2020, Shankar et al., 2021].

Besides, inside-out tracking, all systems have in common that they require a backpack worn by the subject to collect the data. [DuTell et al., 2021] (Figure 2.30 **D)**) states the weight of the headset to be 1.4 kg and the remaining components to be 3.9 kg. Other systems do not report the exact weight, but it can be safely assumed that the headsets are slightly lighter as fewer sensors are used, but the remaining components (Laptop, battery back, etc.) are similar.

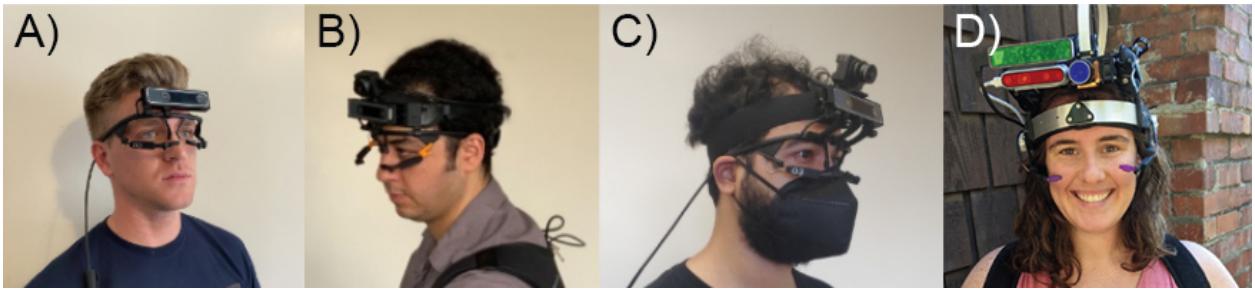


Figure 2.30: Collage of other inside-out tracking approaches. **A)** Positional Head-Eye Tracker [Hausamann et al., 2020], **B)** VEDB headset V1 [Kokhlikyan et al., 2020b], **C)** VEDB headset V2 [Shankar et al., 2021], **D)** High Fidelity Eye, Head and World Tracking [DuTell et al., 2021]. Consents provided as Open Access: Creative Commons Attribution 4.0 International. To view a copy of this license, visit `https://creativecommons.org/licenses/by/4.0/`.

[5]Inside-out tracking describes that the tracking sensor(s) is placed on the tracked body and looks out to determine how its position is changing in relation to the external environment.

With *PESAO* we aimed to equip the subject with as little hardware as possible for multiple reasons:

1. Less slippage for better accuracy
2. Better comfort for more natural movements
3. Faster and easier set up time (including mounting gear, calibrating sensors, etc.)
4. Suited for a wider range of subjects, for instance, headwear, jewelry, etc.

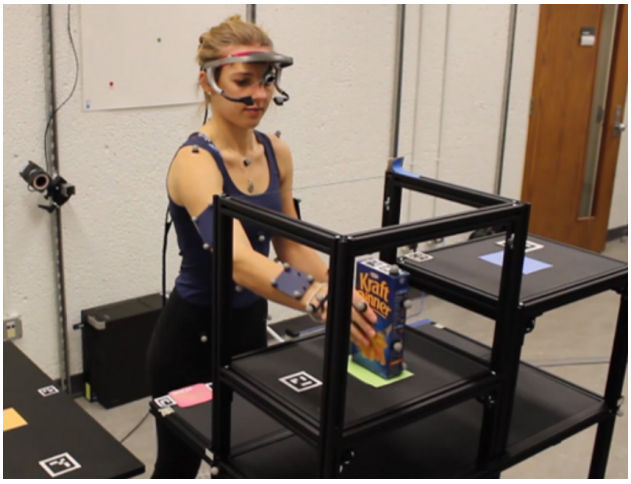


Figure 2.31: Experimental set up of [Stone et al., 2021]. Shown is the *Pasta Box task*. This task consists of three movements: Placing the pasta box from the cart on the first shelf, place it on the second shelf and finally place the box back on the cart. Here a combination of eye tracking glasses and outside-in tracking is used. Similar to *PESAO*. Source: [Stone et al., 2021]. Consent provided as Open Access: Creative Commons Attribution 4.0 International. To view a copy of this license, visit `https://creativecommons.org/licenses/by/4.0/`.

A system that uses a similar set up to track the subject's gaze and movement as used in *PESAO* is presented by [Stone et al., 2021]. Figure 2.31 shows the experimental set up. Notably, a combination of outside-in[6] tracking and eye tracking is used, similar to *PESAO*. Tracking markers are placed on the head, right hand, task cart, side cart, pasta box, and Calibration Wand (calibration). In comparison to the systems presented so far, this is a stationary approach, meaning that the subject needs to stay in a certain position and can only move a few feet.

[6]Outside-in tracking describes that the tracking sensor(s) is placed in the environment and looks in to determine how the tracked body's position is changing.

With *PESAO* we present a system that is easy to set up and calibrate, completely untethered, delivers precise gaze and head tracking, and is truly lightweight – only a pair of glasses with tracking markers and a small $10cm \times 15cm \times 3cm$ processing unit that weighs $\approx 300g$ needs to be worn.

2.3.2 Overview

PESAO is designed for investigating active, visual observation in a three-dimensional world. The goal was to build an experimental set up for various active perception tasks with human subjects (active observers) in mind that is capable of tracking the head and gaze.

While many studies explore human performance, usually, they use line drawings portrayed in two-dimensions, and no active observer is involved [Kjellin et al., 2010, Schaie, 1989, Petrusic et al., 1978, Tkacz-Domb and Yeshurun, 2018, Wloka et al., 2016, Shin et al., 2015]. *PESAO* allows us to bring many studies to the three-dimensional world, even involving active observers. In our instantiation, it spans an area of 400cm x 300cm and can track active observers at a frequency of 120Hz.

Furthermore, *PESAO* provides tracking and recording of 6D head motion, gaze, eye movement-type, first-person video, head-mounted IMU sensor, birds-eye video, and experimenter notes. All are synchronized at microsecond resolution.

In the next sections, we walk through all steps needed to build *PESAO*. We describe the hardware that we have used, how to set it up, and describe PESAOlib, which is the accompanying software of *PESAO*. It is used to design and run an experiment and also to synchronize and analyze the data. PESAOlib is an open-source implementation developed in Python and C++. It provides basic functionalities and is extendable. Figure 2.32 illustrates a sketch of *PESAO*.

In summary, *PESAO* is capable of:

- Head tracking
- Gaze Tracking
- Tracking of Objects
- Controlled Lighting
- Data synchronization

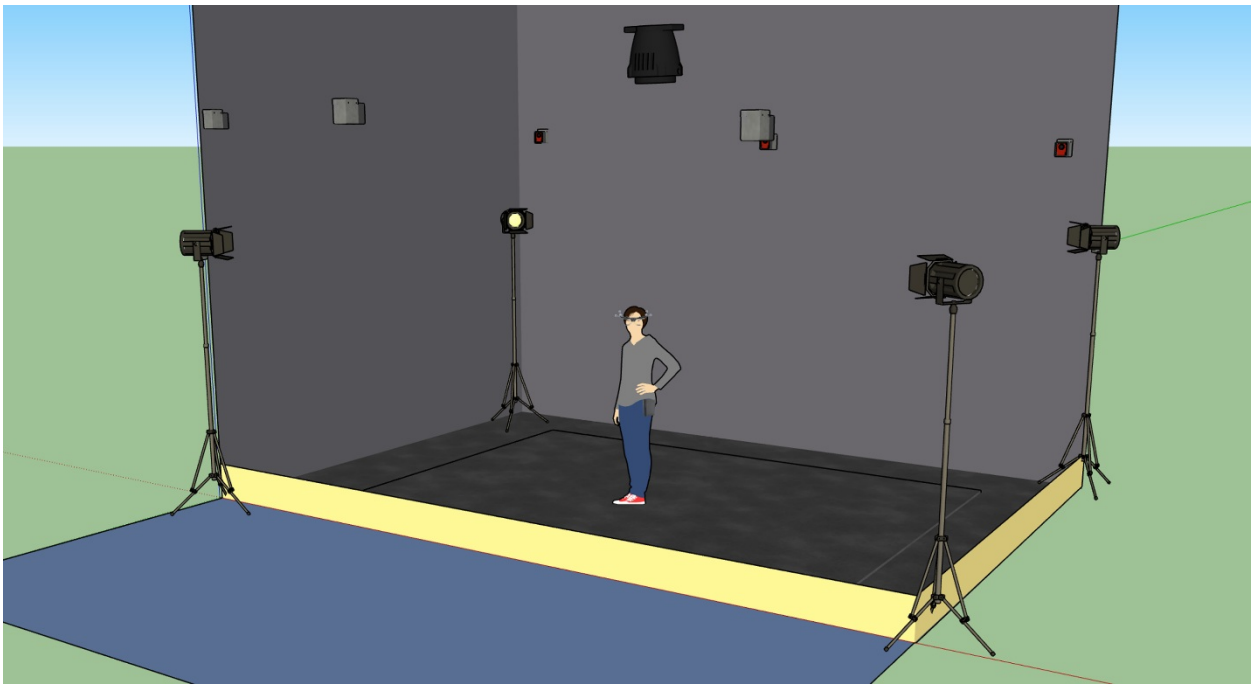


Figure 2.32: A sketch of *PESAO* showing its components: The subject wearing the eye-tracking glasses with motion tracking mount and the compute unit, the five light-sources (one in each corner and one above on the ceiling), six motion tracking cameras.

- Data analysis
- Data visualization

The project page can be found under `http://data.nvision.eecs.yorku.ca/PESAO/`

2.3.3 Hardware

In this section, we describe the hardware that we have used to build *PESAO*. The core hardware pieces are a motion tracking system and an eye-tracking system. Both will be discussed in the following subsections. We will also discuss the glasses' tracking body, which was custom made to track the glasses with the motion tracking system, object tracking bodies, our light-set up and give details on the hardware specifications.

Besides a motion tracking and an eye-tracking system, *PESAO* requires a workstation computer to run PESAOlib.

Based on the hardware configuration, the workstation computer should fulfill the following minimum requirements:

- Windows 10 (requirement of Motive motion capture software)
- > Intel i7-7700k, Ryzen 7 2700x or comparable
- > 8GB RAM
- > 128GB SSD storage
- NVIDIA Quadro, > 2GB VRAM, released 2015 or later (requirement of Tobii Pro Lab software)

2.3.3.1 Motion Tracking System

In this section, we present the motion tracking system we have used in *PESAO*. A range of products exists that are suitable; ultimately, our setup uses a system by OptiTrack[7].

For our implementation of *PESAO*, we chose the Robotics Package with six Flex 13 cameras. With this, up to ten objects can be tracked in a 4m x 4m x 2m volume at 120Hz. This set comes with the most necessary accessories. However, we purchased an additional set of 30 M4 Markers and six 10ft camera stands together with six 3-way head clamps. Figure 2.33 illustrates Flex 13 cameras mounted on stands.

Figure 2.33: Illustration of the motion tracking system with six cameras mounted on tripods. Courtesy of NaturalPoint, Inc., accessed 15 September, 2020, `https://deva90sapmc8w.cloudfront.net/volume12CamStand.jpg`

[7] `https://optitrack.com/systems/#robotics/flex-13/6`

2.3.3.2 Eye Tracking System

For *PESAO*, we chose the Tobii Pro Glasses 2[8] (Figure 2.34).

Figure 2.34: Tobii Pro Glasses 2 Eye Tracking System by Tobii. Courtesy of Tobii AB, accessed 15 September, 2020, `www.tobiipro.com/product-listing/tobii-pro-glasses-2/`

They are capable of recording precise gaze information at either 50 or 100Hz (dependent on the model), first-person video, and motion information (accelerometer and gyroscope). Furthermore, for a more straightforward analysis, we also used the Tobii Pro Lab software to export gaze information and detect eye events. PESAOlib, to be fully functional, requires this data to synchronize the gaze information with the remaining data.

In order to include more subjects in the study and to serve subjects with vision impairment, we purchased the prescription lenses package from Tobii. This package contains corrective snap-on lenses ranging from -5 to +3 diopter in 0.5 diopter steps to facilitate a larger cross-section of subjects, including those with short- or long-sightedness.

2.3.3.3 Glasses Tracking Body

In order to integrate the eye-tracking data with the motion-tracking data, we developed a custom tracking body for the Tobii Pro Glasses 2. The tracking body is a snap-on solution that works with

[8] `https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/`

most hairstyles and head decorations. We put a focus on reliability, durability and modularity in order to never lose a subject to failing hardware or lose their data. The tracking body has a modular design for a piece-wise exchange of broken elements. Figure 2.35 shows the custom tracking body including a exploded view drawing (bottom right).

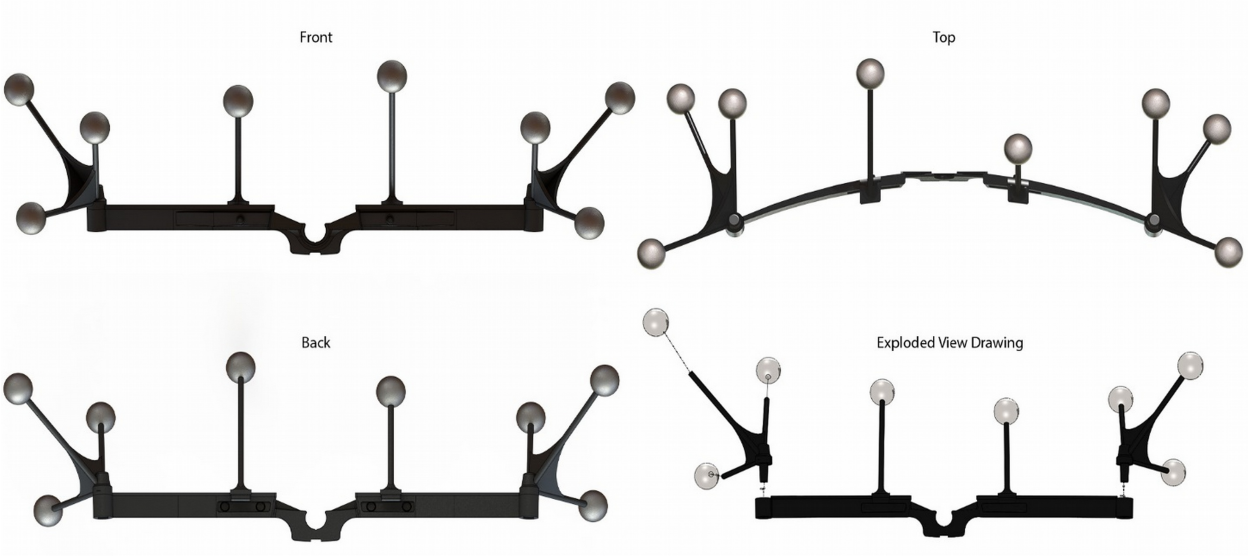


Figure 2.35: Tobii Pro Glasses 2 Custom Tracking Body from different angles and with exploded view drawing (bottom right). We used a modular design that allows to replace parts instead of the entire unit if something needs to be adjusted or breaks.

The tracking body is equipped with standard M4 Markers (available through OptiTrack). If it is required to use prescription lenses with the system, it is necessary to attach four magnets to the tracking body (⌀4mm). The 3D printable file for the tracking body and print settings can be found in the README file on the project page.

An assembled tracking body mounted on Tobii Pro Glasses 2 can be seen in Figure 2.36. The tracking body does not change the tracking capabilities of the glasses nor obstruct the field of view of the subject. However, it is necessary to set up and calibrate the tracking body in the Motive motion tracking software. As a result, *PESAO* records precise six-degree-of-freedom tracking information of the tracking body.

While conducting experiments, the visibility of the first-person camera of the glasses positioned between the eyes of the subject should be checked. Subjects tend to push the glasses up their nose by pressing against the bridge of the glasses where the camera is positioned, hence smearing the camera lens. The glasses and camera lens should be regularly cleaned.

Figure 2.36: The tracking body mounted on the eye-tracking glasses.

2.3.3.4 Object Tracking System

For several visual perception experiments, it is also of interest to track the position of the object under investigation.

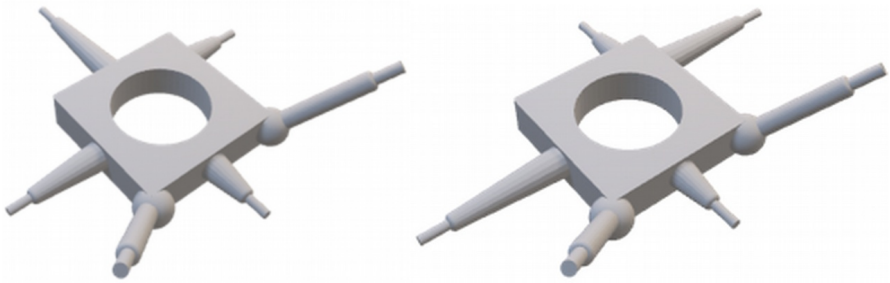


Figure 2.37: A pair of tracking bodies to track the position of objects within *PESAO*. It was important to us to design the bodies that it is visible even if the subject occludes most of it, hence we used a rotation variant layout of size tracking markers distributed over all four sides.

We have designed two custom object tracking bodies that can be used to track objects in three-dimensional space. The bodies can be either mounted directly on the object or on a rod with a 2.54cm diameter. The objects are designed for tracking reliability and ease of use.

Similar to the glasses tracking body, the object tracking bodies are available as a 3D printable file. Figure 2.37 presents an illustration. For further information on print settings, please review the README file on the project page. The tracking bodies have to be calibrated in the Motive motion tracking software in order to be recognized by *PESAO*.

2.3.3.5 Light

Our visual system cannot function without light; hence, to study the effect of light, *PESAO* offers controllable light settings and light measuring capabilities. We have used 660 LED Video light panels from Neewer[9]. They are offered with stands. An essential feature of these lights is their capability to dim and change the colour temperature.

For our tracking area of 400 x 300cm, we have used five light sources, one in each corner and one above functioning as a ceiling light. See Figure 2.32 for a sketch of the set up, including the positioning of the light sources.

The light panels provide colour temperatures from 3200 – 5600K and lumen of up to 7300 Lux/m.

2.3.3.6 Hardware Specifications

The hardware used for *PESAO* requires different parameters; hence, the experimenter can alter possible independent variables to test the effects on dependent variables of the study. In Table 2.3, we provide a list of hardware parameters, their possible value range, and if, applicable, their accuracy.

2.3.4 Hardware Set Up

After going over the necessary hardware for PESAO, we give a brief overview of how to put all the pieces of hardware together. In addition, we also included a birds-eye camera to record subjects from above (Figure 2.38). In order to do this, almost any available webcam can be used.

Figure 2.38 gives an overview of how to integrate the hardware. Important to note is that PESAO relies on two different connectivity standards; USB 2.0[10] and WiFi 5[11] (IEEE 802.11ac).

[9] `https://neewer.com/collections/led-panel-light/products/nl660-led-panel-lights-90095562`
[10] `https://www.usb.org/`
[11] `https://www.wi-fi.org/`

Hardware	Parameter	Value Range	Accuracy
Eye-Tracking	Tacking Frequency	50 or 100 Hz	
	IMU Frequency	50 or 100 Hz	0.05° static / 0.08° dynamic
	Gaze Tracking	50 or 100 Hz	1.42° Mean Accuracy
	Eye movement type	50 or 100 Hz	
	First-Person Camera	25 FPS	
Motion-Tracking	Tacking Frequency	120 Hz	0.2mm (97% capture volume)
Light	Light Intensity	7300 Lux/m	
	Colour Temperature	3200-5600 K	
Camera	Frequency	30Hz	

Table 2.3: *PESAO* Hardware Specifications

Components able to be connected over USB should be connected over a wired connection for robustness. However, the eye-tracking system, to allow unconstrained experiments, is connected over WiFi.

The WiFi connection is used to control and live view the eye-tracking system. Specifically, it is used to set up the eye-tracking glasses, start and stop the data recording, and acquire frequent synchronization timestamps of the system. The recording itself (gaze information, first-person video, calibration data) is stored on an SD-card in the recording unit of the eye-tracking system and will be copied after the experiment. Tobii includes with the Tobii Pro Glasses 2 API the capability of directly recording gaze data over WiFi, but in experiments, we have found a recording on the internal SD-card yields better tracking results.

Figure 2.39 shows an example set up of PESAO with dimensions. Important to note is that between the tracking and control area, a blackout curtain restricts the view so that the experimenter's movements do not distract the subject.

2.3.5 PESAOlib

PESAOlib is designed to control and execute experiments, record data with precise, accurate to microsecond-level timestamps, as well as synchronize and analyze the recorded data. PESAOlib provides a comma-separated values file (CSV) and a pickle file as output.

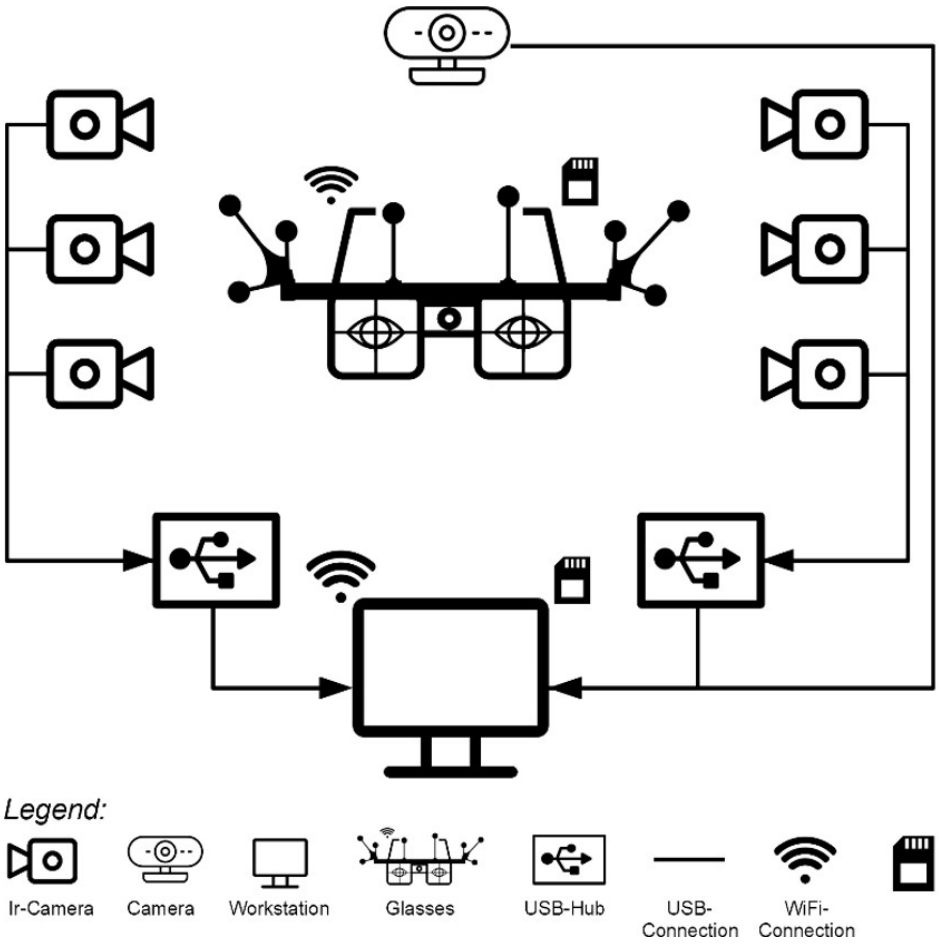


Figure 2.38: *PESAO* hardware connectivity overview with all components: motion tracking camera, camera, workstation computer, eye-tracking glasses, USB-Hubs, USB-Cables, WiFi-Connections and SD-Storage.

2.3.5.1 Overview

The software is written in Python and C++ and uses the networking, and synchronization functionalities of the well-established lab-streaming layer [Kothe, 2014] by Swartz Center for Computational Neuroscience.

We provide several source codes that are ready for compilation under Windows 10. Software parts that might vary from experimental design to another experimental design are provided in Python to be easier adjustable for programming novices.

Figure 2.40 displays a diagram showing the dependencies between PESAO's modules. All programs are executed on the Workstation (Figure 2.38). However, as PESAO is designed, programs can be run on multiple, connected workstations. This might be of interest if more sensors or a higher

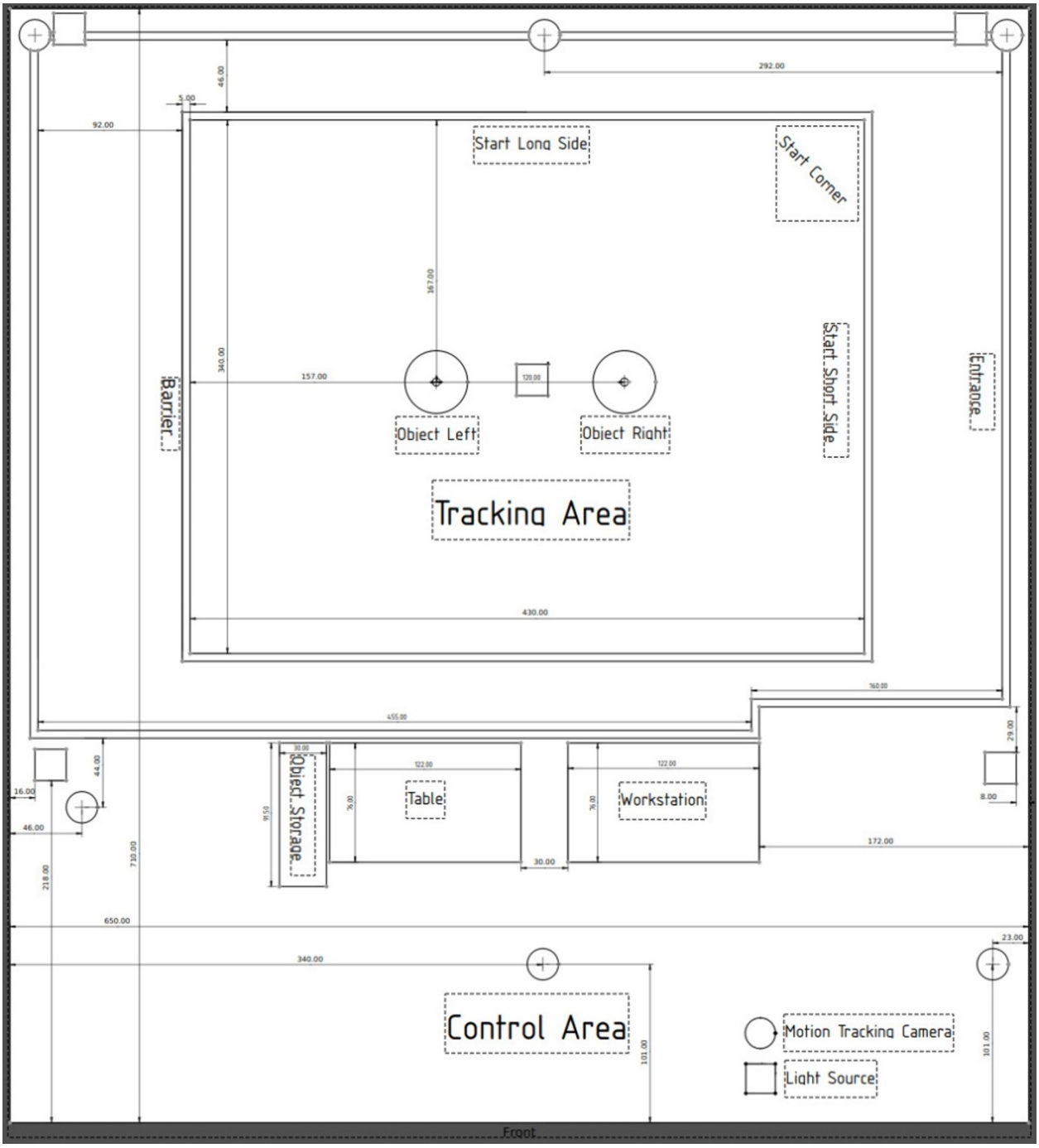


Figure 2.39: *PESAO* dimensions as used in this thesis. Different areas are illustrated in which the subject performs the experiment (tracking area) and observe, control, and record it (control area). Not illustrated in the drawing is a visibilty barrier running between tracking and control area, so the subject does not get distracted by the investigator.

bandwidth is needed.

PESAOlib creates and saves files along its processing pipeline to help the user to understand

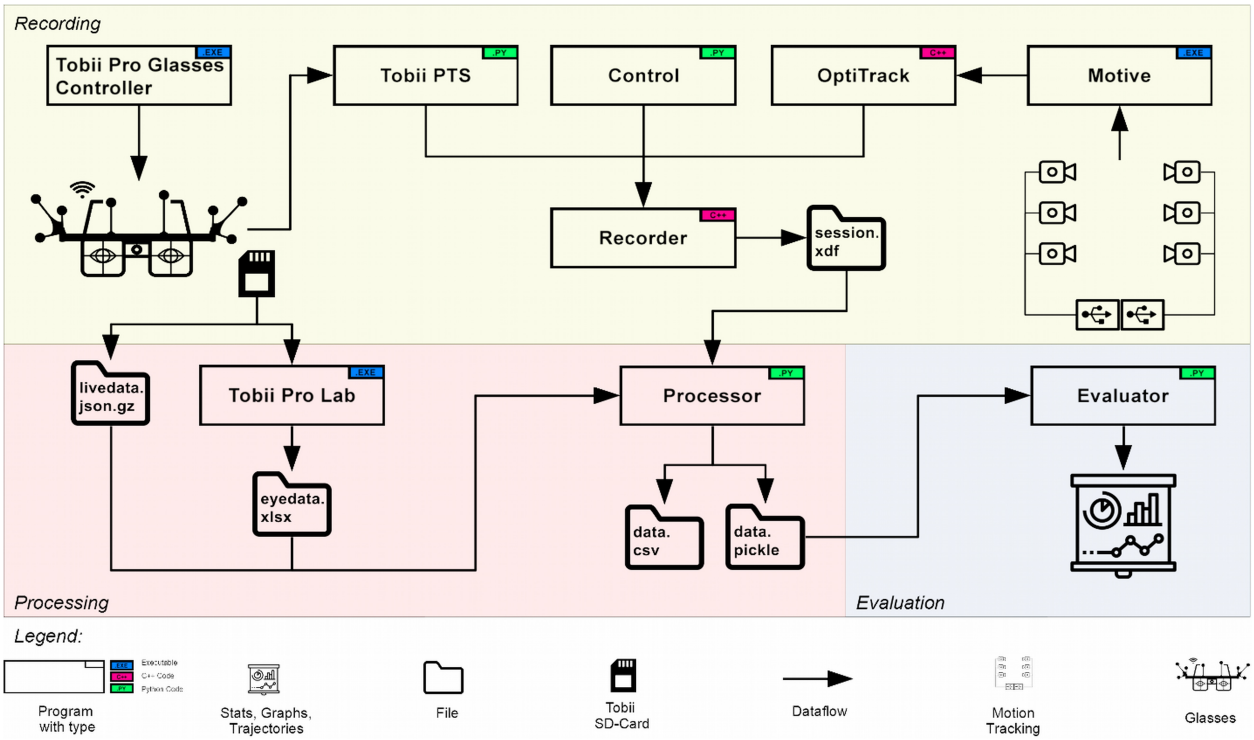


Figure 2.40: *PESAOlib* overview with all its modules, dependencies and outputs. *PESAOlib* is divided into three main parts: recording, processing, and evaluation. All software required from running an experiment to create many different visualizations can be found here.

intermediate steps better and modularize their workflow. A strength of this system is that the subject is entirely untethered, allowing free and natural motion. To compensate for transmission loss and lag, the eye-tracking device records on its local SD-Card.

As mentioned earlier, adjustable PEASAOlib modules are written in Python (green type indicator). However, modules that require high performance and are foreseen to remain unchanged for most set up scenarios are implemented in C++ (magenta type indicator). Nevertheless, the C++ source code is provided and can be changed as desired. Lastly, to allow the best interoperability with the physical devices, such as eye-tracking and motion-tracking systems, we use the programs provided by the manufacturer (blue type indicator).

2.3.5.2 Recording Module

Figure 2.40 top (green) depicts the Recording section of PESAOlib. These are modules responsible for gathering the data, including the subject's gaze and head motion, synchronization timestamps, and experiment instructions, notes and feedback from the subject.

Starting on the top left, the executable Tobii Pro Glasses Controller controls, as the name suggests, the Tobii Pro Glasses II. With this program, you can set the subject's name, check the battery and storage status of the glasses, calibrate the glasses, and start and stop recording. It is straightforward to use. This includes an integrated update function to update the glasses' firmware.

Tobii PTS is a program written in Python that functions as a command-line program without a graphical user interface. Its task is to acquire recording timestamps from the glasses, which are used to synchronize the motion tracking system and the control program with the glasses' data. The program can be started at any point before the experiment, and it will begin collecting timestamps once the glasses are recording (initiated in Tobii Pro Glasses Controller).

Similarly to the Tobii PTS program, Control is a Python command-line program. The task of this tool is to assist the experimenter with walking the subject through the trials, generating randomized trials, taking observational notes, and recording the subject's answers. This Python program is your starting point to design a new experiment.

For performance reasons, we implemented the program that gathers the live-streamed motion-tracking data in C++. With PESAO, we supply an executable that works with most OptiTrack cameras and also the source code. Figure 2.41 shows a screenshot of the user interface. The program also allows you to change the camera type, set frames per second, and a number of network settings to connect to Motive. However, the default network settings should suffice if Motive is run in default.

In order to calibrate and run motion tracking, PESAO relies on OptiTrack's Motive program. Motive is supplied with your OptiTrack Motion Tracking system — the program interfaces with tracking cameras and tracks the defined bodies. With PESAO, we supply a 3D head-model-file and 3D model files of the object tracking bodies for visualization. Make sure to have Motive's live-streaming function enabled. Otherwise, OptiTrack will not be able to link to it. We recommend calibrating the motion tracking system at least once a day, and every time a tracking camera is moved. Figure 2.42 shows a screenshot of the user interface. Once the live-streaming functionalities have been enabled, all that is needed is to start Motive. It will automatically load the latest calibration, trackable bodies and immediately start live-streaming.

The program that wraps everything together and is responsible for the recording is the *Recorder*. This program is implemented in C++ and comes with a user inface. See Figure 2.43.

Following the startup of the previously mentioned programs, the Recorder is the last pro-

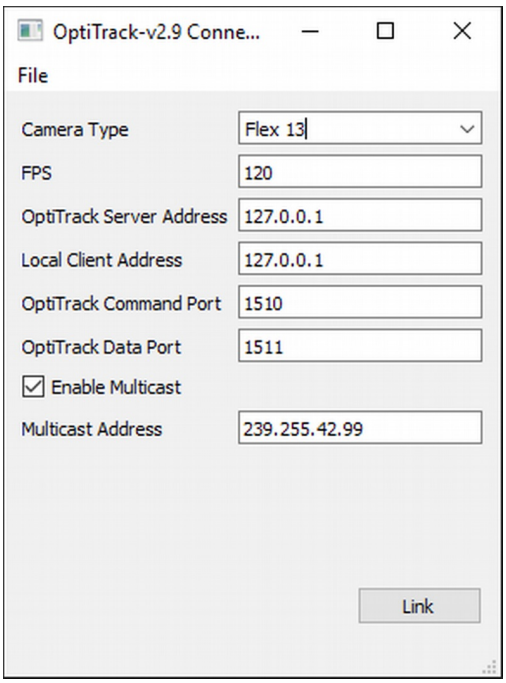


Figure 2.41: *PESAO* OptiTrack Module. This module connects and collects data from the motion tracking system. This module has been largely taken from the lab-streaming layer example repository.

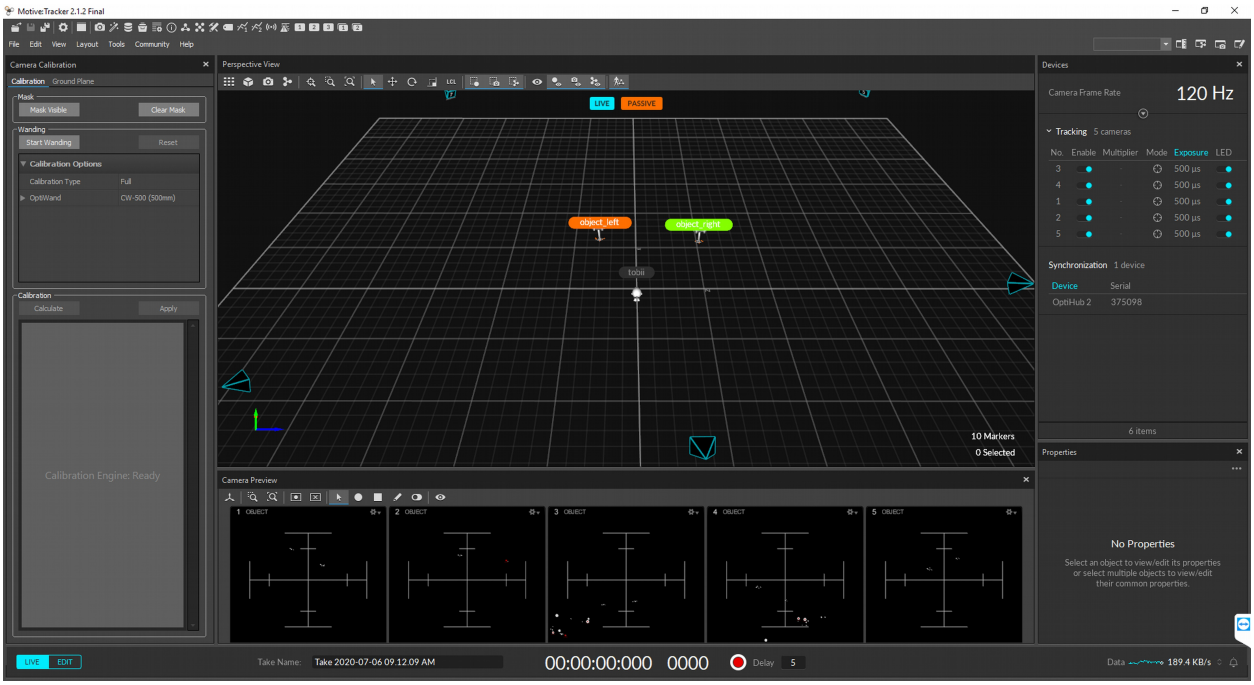


Figure 2.42: OptiTrack's Motive user interface. This software is used to calibrate the motion tracking volume and to set up rigid bodies, such as the tracking body mounted on the eye-tracking glasses and the tracking bodies.

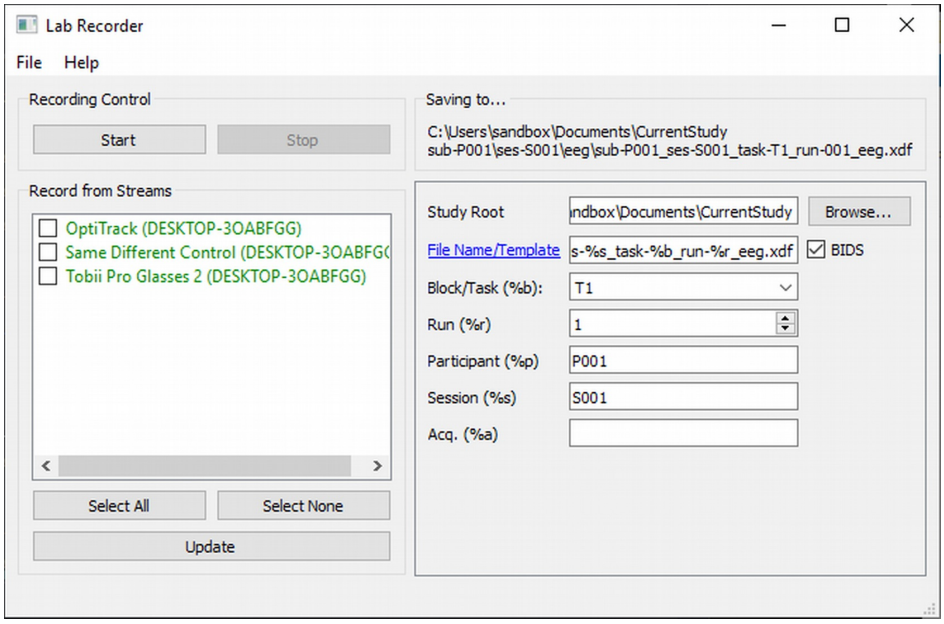


Figure 2.43: The Lab Recorder module is the central piece of software to record the data from all other lab-streaming layer modules. In this version, it is connected to three other sources: the motion tracking system (OptiTrack), control script of the experiment (Same Different Control), and the eye-tracking glasses (Tobii Pro Glasses 2).

gram needed to record an experiment. The user interface provides information about the available streams, and it should show three streams; OptiTrack, Control, and Tobii PTS. If the stream does not show and you are certain that all programs are running, press the Update button to search for available streams. On the right-hand side, you are able to set parameters for the experiment and subject under investigation. The data will be saved into an XDF format in a folder specified under Study Root. To start recording, select all streams and press the Start button on the top left. To stop the recording, press Stop. With this program, we conclude the Recording section of PESAOlib.

2.3.5.3 Processing Module

After successfully recording the experiment, PESAOlib provides a processing program that cleans and synchronizes the motion tracking data with the control data and the gaze data. This section is highlighted in a light red.

Figure 2.40 shows that the processor needs three files; the session.xdf produced by the Recorder, the livedata.json.gz, which can be loaded directly from the SD-Card of the glasses' recording unit

and the eyedata.xlsx file. The eyedata.xlsx file has to be created by the analysis software Tobii Pro Lab. A screenshot of this software is displayed in Figure 2.44. Make sure to select the "Single Excel file (.xlsx)" as a setting for the format and select all data to be exported. The software provides more options to filter the eye-tracking data. These can be adjusted as needed.

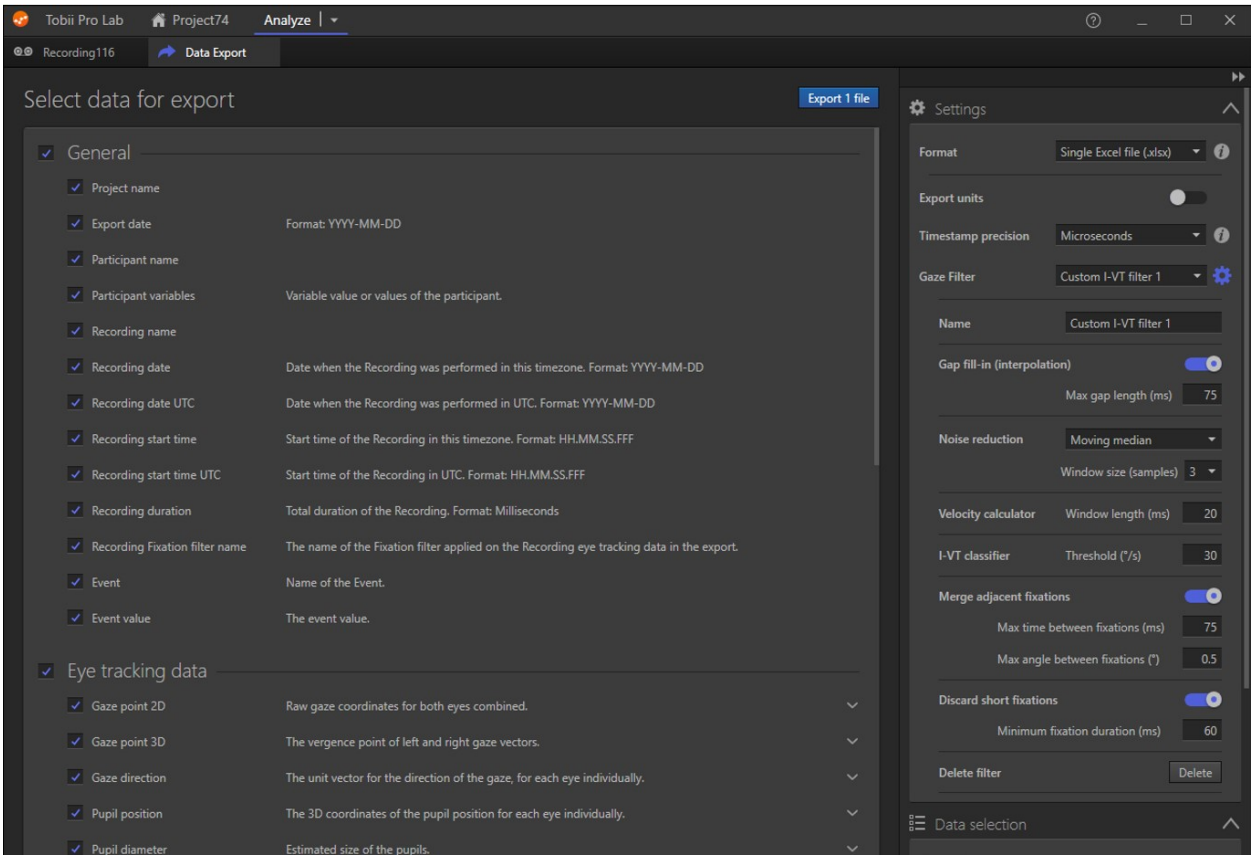


Figure 2.44: Tobii Pro Lab export dialog. Showing the preferred settings for the use with *PESAO* (right-hand side). This step will export an excel spreadsheet with detailed eye-tracking information.

2.3.5.4 Evaluation Module

Lastly, PESAO provides with the PESAOlib Evaluator (Figure 2.40, light blue area) tools and examples to evaluate the generated data. *PESAO* was developed so that each process can be easily understood and examined. To realize this, besides providing source codes and developing PESAOlib mainly in Python, most of the generated artifacts are readable by humans (non-binary files) or a human-readable file is supplied alongside the binary file (e.g. CSV file).

The evaluation section consists of the Evaluator program written in Python. The program

supplies multiple utility classes to filter through data.pickle, plot graphs, annotate your plots with 3D STL models and more. Furthermore, the Evaluator program supplies you with a set of examples. One such example can be seen in Figure 2.45. It visualizes viewing frusta in temporal colour coding of subjects performing a visual task.

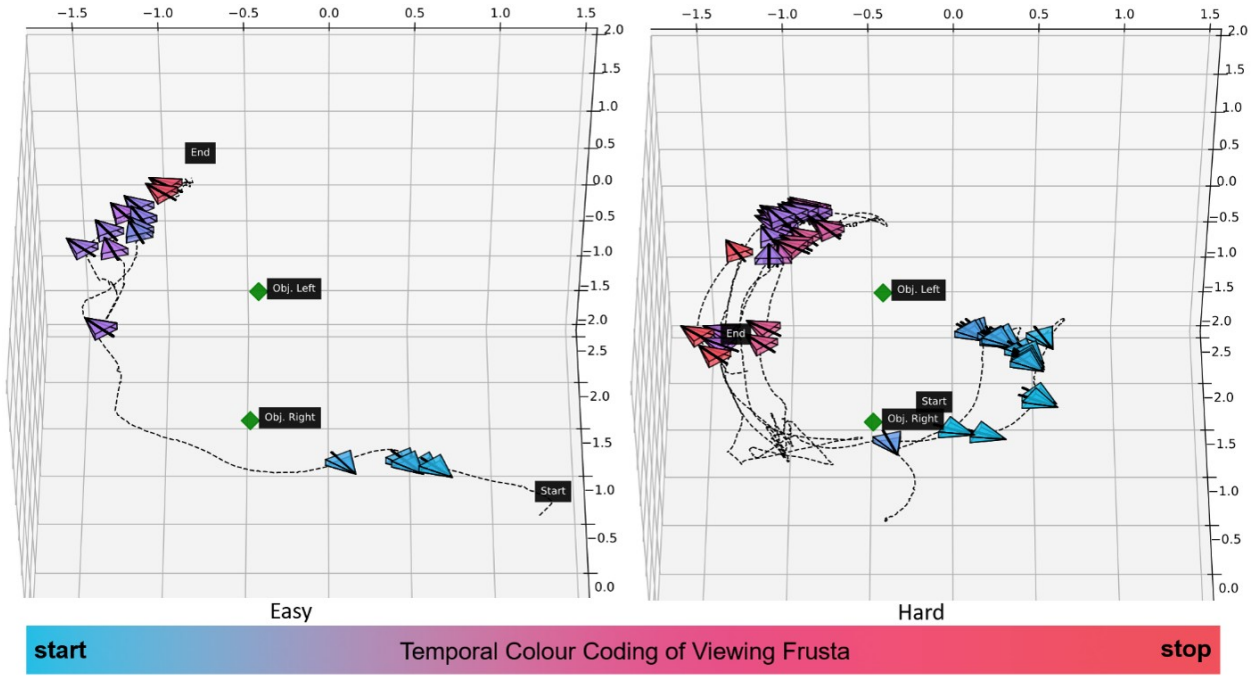


Figure 2.45: Example visualization using *PESAOlib* Evaluator. Here two trials are plotted (left and right). Specifically, the trajectory of the head movement (dotted line), fixations (viewing frusta), and position of two objects. The viewing frusta are shown in temporal colour coding from blue (start) to stop (orange). Furthermore, the start and endpoints of the trajectory are annotated with corresponding labels.

2.3.5.5 Project Page

In conclusion, PESAOlib provides you with three modules that cover everything from recording to processing and evaluation. Technical details can be found in the code documentation and check for updates for PESAOlib. PEASOlib is still in active development. To obtain a copy, please visit the GitLab home at `https://gitlab.nvision.eecs.yorku.ca/solbach/pesaolib/`. A screenshot of the project page is shown in Figure 2.46.

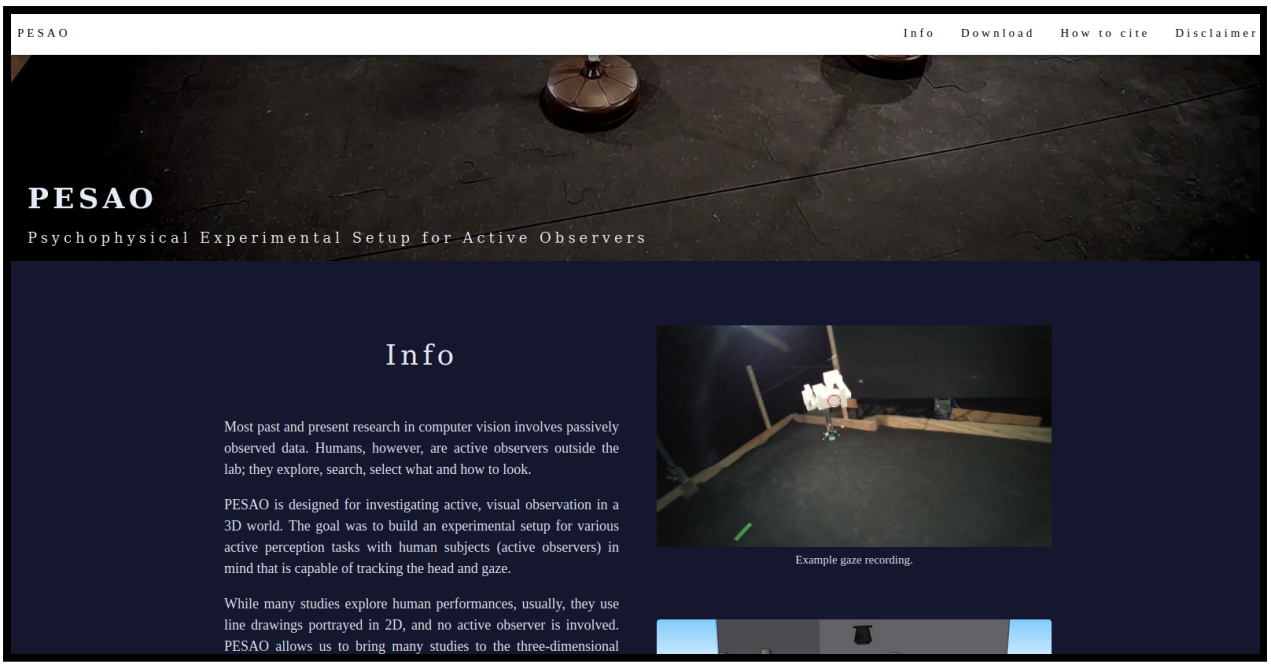


Figure 2.46: Screenshot of the *PESAO* project page. The page can be found under `https://gitlab.nvision.eecs.yorku.ca/solbach/pesaolib/`. Further information about *PESAO* can be found there.

2.3.6 Future Work

In conclusion, PESAO provides precise and synchronized head and eye-tracking that allows for the investigation of active visual observation in a three-dimensional world. Mainly building upon open source software, we hope it finds many use-cases and different instantiations in and outside laboratories worldwide.

In the future, we hope to extend PESAO to different hardware vendors, making it accessible cross-platform, and perhaps integrate more sensors, such as full-body tracking, including legs and hands, as well as light sensors on the stimulus to investigate shadowing. Further, new lightweight and portable EEG headset with C/C++ SDK, like the one from Bitbrain as shown in Figure 2.47, would also be a beneficial integration of EEG data in *PESAO*.

Having presented our experimental facility and introduced a novel set of objects, in the next section, we explain how we run the *Three-Dimensional Same-Different Task for Active Observers*.

Figure 2.47: A possible extension for *PESAO*: EEG device from Bitbrain could be worn simultaneously with the eye tracker. Courtesy of Bitbrain, accessed 11 March, 2022, `https://www.bitbrain.com/neurotechnology-products/dry-eeg/diadem`

2.4 Experiment Design

The visuospatial task posed to human subjects is to determine whether two presented three-dimensional objects are the same or different. It is permitted to move around and change the position to gather different viewpoints of the objects. However, it is not permitted to touch or manipulate objects.

2.4.1 Ecological Validity

The classic instance of the same-different task is widely known from the work of [Shepard and Metzler, 1971]. There, they used objects formed by concatenations of cubes and depicted as black line perspective drawings on a white background. Subjects were shown pairs of these objects and were asked if the objects were the same or different. Stimuli were 4-5cm in linear extent, seen in two windows, viewed from 60cm. For an example, see Figure 2.6.

In other words, subjects were passive viewers with a constant target visual angle for each stimulus object. The "view" was pre-determined. Since reaction times were as long as 5s, there was plenty of time for eye movements, but no report of them was provided. Results showed that subjects seemed to mentally rotate one object into the other, this being an inference made by considering response time. However, something important is missing here.

Humans did not evolve categorizing two-dimensional line drawings on a screen; thus, we sought to move this classic study into a more ecologically valid setting. We push this experiment to its

limits, increase stimulus complexity, perform it in three dimensions, allowing observers to choose whatever viewpoints they wished towards solving the task and record in detail how they viewed the objects. In this way, we wanted to discover exactly how a human agent solves such a real three-dimensional visuospatial task.

2.4.2 Explaining the Task to the Subjects

When dealing with human subjects, it is vital to clearly lay out what the participant is asked to do and avoid biases due to miscommunications.

Subjects are asked to wear a pair of eye-tracking glasses that require calibration before the trials start. The calibration will be done after the explanation and takes around 5-10 seconds. After successful calibration of the system, the subject is asked to approach the first starting point.

After a trial ends, the subject approaches a randomly assigned starting point facing the curtain; the experimenter enters the tracking area and changes the objects accordingly to the directions of the control application (see Section 2.4.3). Before the first trial can start, the subject has to sign the University's consent form (a copy is provided in Appendix A). After signing, the experimenter explains the study to the subject following these bullet points:

- It is possible to withdraw your participation at any time, especially if any form of discomfort is experienced.
- You will perform a total of 18 trials.
- It is up to you how much time you take per trial.
- This is a forced-choice experiment. Meaning that you need to answer with "same" or "different" to end the trial.
- You will be presented with two objects.
- Your task is to determine if objects are the same or different. The same means that they have the same appearance (geometry, size, colour). Different means that the geometry is different.
- It is allowed to move around freely within the boundaries (see Section 2.3 for details)

- Try to answer as quickly as possible but only answer if confident (an additional view is more valuable than a wrong answer).
- If a decision is made, provide the experimenter with your answer: "same" or "different".
- You are presented with a starting point (see Section 2.3 for details) and asked to stand there facing the black curtain.
- When the experimenter says, "begin," you may turn to face the objects and begin the next trial.

2.4.3 Control Design

A control application is implemented that allows us to collect the name, answers and observation notes for each trial with a global timestamp. Collected data is synchronized with the tracking information, which allows us to calculate the duration of each trial and more. Furthermore, the control application randomly defines the order in which the 18 objects are presented, the orientational difference of the objects, the starting positions of the subject and whether the objects are the same or different.

2.4.4 The Stimulus

For the three-dimensional version of the same-different task, we use the *TEOS* L_2 objects as a stimulus as presented in Section 2.2. We choose the L_2 and not the L_1 set, as we need a high intra-class similarity to push this experiment to the limits. In pilot studies, we found that a low intra-class similarity resulted in simply counting the elements which make up the object. An example object from three different views is shown in Figure 2.5.

2.4.5 Object Rotation

To investigate the role of object orientation, we will be presenting the objects with different orientations with respect to a global coordinate system. Of great interest in this task is to investigate which role the orientational difference between both objects has. For this, we need first to specify what we mean by this.

Since the configuration space of possible orientations is very large, we quantize the viewing sphere to reduce the number of possible orientations. For example, even if we are using a rather big stepping distance of 10° for elevation and azimuth, plus taking into account that we deal with two objects at the same time that can be altered, we have in total

$$\left(\frac{360}{10} \times \frac{360}{10}\right)^2 = 1,679,616 \tag{2.3}$$

possible orientational differences. This number is beyond the scope of this research. Therefore, we quantized the viewing sphere around an object into eight evenly distributed viewing vectors v_1 to v_8 (see Figure 2.48). Each vector stands for one possible pose of the object. Due to the use of axis angles, the Y-vector of the object is aligned with a viewing vector by only rotating the object around its X- and Z-axes. Figure 2.16 shows the local coordinate system of the object. The local coordinate system is expressed with respect to its viewing sphere, which in turn is expressed with respect to the global coordinate system of the tracking area (more information in Chapter 2.3.

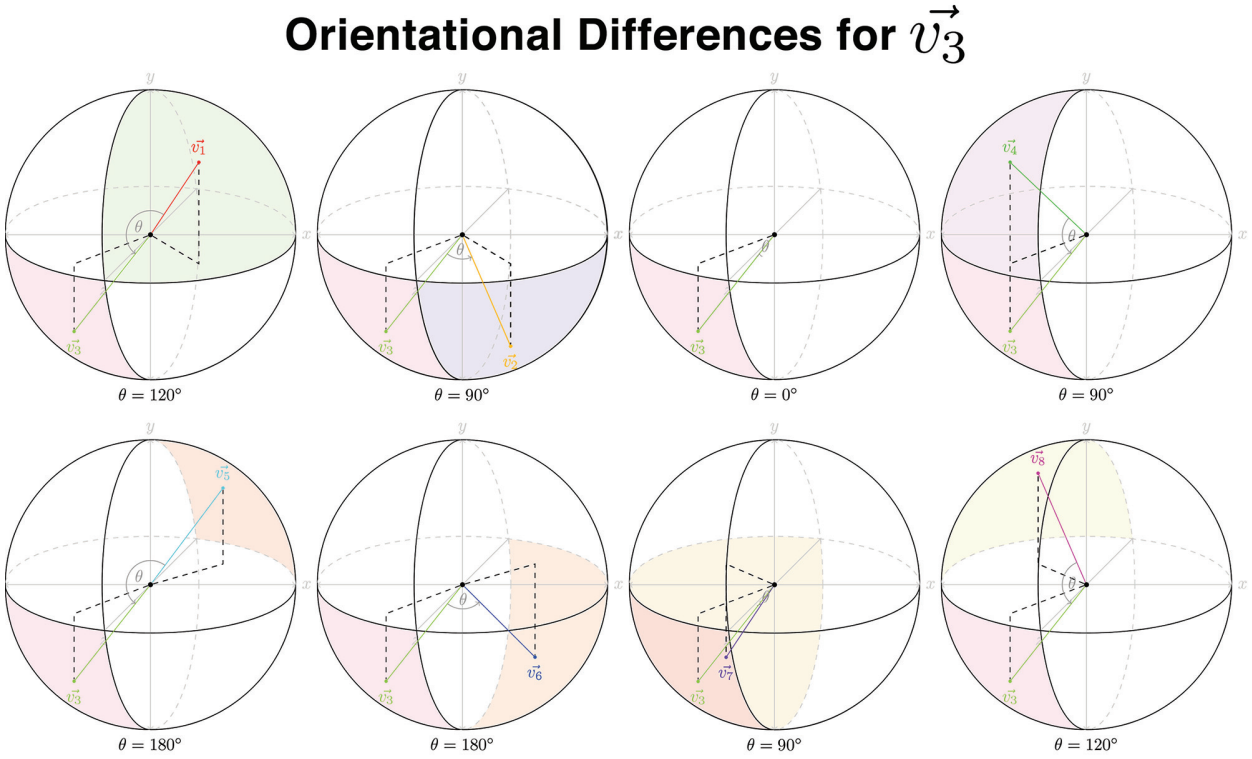


Figure 2.48: Octahedron viewpoint projection and orientational differences of v_3.

However, using eight different viewpoints for each object still gives 64 possibilities to orient

both objects to each other.

The 3D rotation group $SO(3)$ admits several representations that can be converted into each other. Most simple, a viewpoint vector can be expressed as a rotation in axis-angle, which we will be converting into a rotation matrix representation next. However, independent of the representation, there is an intrinsic notion of distance on $SO(3)$.

In short, to calculate the orientational difference between two viewpoints, we have to retrieve the angle from the different rotations from the trace of R, where R is the distance between rotations [Trefethen and Bau III, 1997].

Let $R_{v_y}^T$ denote the matrix transpose of R_{v_y}. The difference rotation matrix D is defined as

$$D = R_{v_x} R_{v_y}^T. \tag{2.4}$$

Next, the distance between rotations represented by R_{v_x} and R_{v_y} is the angle of the difference rotation represented by the rotation matrix D. The angle of the difference rotation can be retrieved from the trace of D:

$$\theta = \arccos \frac{\mathrm{tr}(D) - 1}{2} \tag{2.5}$$

Using this, we can see that the orientational differences for some of the viewpoints are the same. An example is given in Figure 2.48 in which the orientational differences for v_3 against all other possible viewpoints, including itself, are illustrated.

For this experiment, we will choose three orientational differences to reduce the configuration space of the experiment. Under investigation will be the 0° distance of a given viewpoint to itself (Figure 2.48 v_3 to v_3), 90° distance (Figure 2.48 v_3 to v_4) and 180° distance (Figure 2.48 v_3 to v_5).

2.4.6 Starting Position

It is believed that the starting position may play a vital role in how the subject will perform the task. The starting position defines the initial view of the objects and is, therefore, crucial for the decision process of upcoming views.

In total, three starting positions are under investigation that covers the possibilities of seeing

both objects next to each other (Figure 2.39: Start Long Side), form an angle (Figure 2.39: Start Corner), and one object is occluded by the other (Figure 2.39: Start Short Side).

2.4.7 Experimenter Effects

The *Experimenter Effects* refer to the experimental artifact that the subject consciously or unconsciously aims to produce the results that meet what they think are the expectations of the experimenter.

It is not foreseen that the experimenter will influence the participant throughout the study as no feedback during and after trial completion is given. Further, the subject will perform the experiment autonomously after receiving the start signal from the experimenter.

However, it is possible for the experimenter to intervene if it is believed that instructions were unclear; for instance, if the subject does not take enough time to assess the objects and guess the answer – there would be an unusually high occurrence of wrong answers. Similarly, this can be monitored by checking whether or not the collected dependent variables (Section 2.4.9) vary significantly from what has been observed with other subjects.

2.4.8 Demand Characteristics

The *Demand Characteristics* describes unintentional hints and cues to the subject provided either by the investigator or the environment that bias the research findings.

As described in Experimenter Effects, it is not planned to influence the subject throughout the experiment at any time, except to tell the subject to start the trial and, if needed, to reiterate the instructions. Therefore, it is not foreseen that the experimental design will unintentionally provide subjects with hints.

2.4.9 Experimental Variables

In this section, we present different types of experimental variables that affect the experiment. It is necessary to clarify them for the reproducibility of the investigation. In total, we define four variables; independent, dependent, extraneous, and confounding variables. Further, we describe our method of random allocation and address the potential of order effects.

Independent Variables

Independent Variables are variables that are manipulated in the study to explore their effects. These variables are called "independent" as they are not influenced by any other variables in the study.

The experimenter will alter in total four out of nine *Independent variables*. Table 2.4 presents all variables and their definitions.

Dependent Variables

Dependent Variables change as a result of the manipulation of the independent variables. They are also known as response variables as they respond to a change in another (independent) variable. The measured data of dependent variables describes to what extent independent variables influence dependent variables by conducting statistical analysis.

The experimenter will measure eight *Dependent Variables*. Table 2.5 presents all variables, type of data and their frequency.

Extraneous Variables

Extraneous Variables are variables that are not *Independent Variables* but could affect *Dependent Variables*. In other words, these are any variables that are not investigated that can potentially affect the outcomes of the study. Table 2.6 presents all variables and the proposed measurements to deal with them.

If any of the above extraneous variables occur, it needs to be considered to restart the trial to avoid the collection of negatively affected data.

Confounding Variables

Confounding Variables are a type of extraneous variables which are not controlled and which influence both the independent and dependent variables. These variables usually result in the termination of the experiment. Table 2.7 presents the confounding variables of the variable for this experiment.

Random Allocation

Random allocation describes how participants are allocated to different experimental "versions", or in this case, independent variables. For this, we use a control application to guarantee that participants are randomly allocated to *Independent Variable* conditions.

The entire configuration space of this experiment is

$$com \times sd \times rot \times pos = 3 \times 2 \times 3 \times 3 = 54, \tag{2.6}$$

where *com* stands for the levels of complexity of the objects, *sd* stands for whether the objects are the same or different, *rot* stands for the number of possible rotations and *pos* stands for the number of possible starting positions. To cover the entire configuration space, at least 54 trials have to be performed. Since every subject is asked to do 18 trials, three subjects are needed to sample the entire configuration once. [Belke and Meyer, 2002] stated that sampling the configuration space more than 15 times for the same-different task is preferred. This leads to a group of 45 subjects.

Order Effects

The *order effect* describes the differences in responses that result from the order in which the independent variables are presented.

Every subject takes part in eighteen trials, roughly divided 50:50 into same and different. However, the order in which same and different objects are presented, as well as all other changing independent variables, is completely randomized (see *Random Allocation*. Hence, we do not expect any order effect.

2.5 Summary

In this chapter, we have presented the *Three-Dimensional Same-Different Task for Active Observers*, including a description of the experiment for reproducibility. The same-different task has already seen many implementations in different fields – however, none in three-dimensions, including active observation. We have laid the foundation to use this task in a laboratory environment providing, as well as introduced a novel set of objects, an experimental set up to record and analyse human visuospatial problem-solving.

In the next Chapter 3 we present our experimental results of running hundreds of different trials of the *Three-Dimensional Same-Different Task for Active Observers*.

Independent Variables	Definition
Object	24 physical, 3D printed objects. Twelve objects (Figure 2.5) with a corresponding clone split into three complexity levels (easy, medium, hard).
Object Pairing	Either an object and its clone (same) or an object with another object from the same class (different) to avoid merely counting the number of cuboids.
Rotation	We investigate in total three different orientational differences: 0°, 90° and 180°. For further information, please see section Object Rotation.
Starting Position	The starting position of the subject: short side, corner, long side. For further information, please see section Starting Position.
Light Source Number	Five light sources. Four LED and one Halogen light source. All are equipped with diffuser panels for even lighting. This variable will remain unchanged in this experiment.
Light Source Position	The LED light sources are positioned in each corner of the tracking area at 1.9m height. The Halogen light source is positioned above the objects at 2.5m in height. This variable will remain unchanged in this experiment.
Light Source Direction	All light sources are facing the objects. This variable will remain unchanged in this experiment.
Light Source Intensity	The LED light sources operate at 3360 Lux/meter. The Halogen light source operates at 9000 Lux/meter. This variable will remain unchanged in this experiment.
Light Source Color Temperature	The LED light sources operate at 3200K (Cool White), and the Halogen light source operates at 2500K (Warm White). This variable will remain unchanged in this experiment.

Table 2.4: List of *Independent Variables* and their definitions.

Dependent Variables	Type	Frequency
Same-Different decision	Verbal answer	Once a trial
Time taken for trial	Seconds	Once a trial
Set of viewpoints	6D Pose	120Hz
Trajectory of subject's head	6D Pose	120Hz
Gaze	x-, $y-$ coordinates on 2D image-plane	50Hz
Eye movement type	Fixation, Saccade, PointTrackingGaze, EyesNotFound	50Hz
Camera Feed	Outward-facing camera feed	25Hz
Camera Feed II	Stationary camera recording the tracking area	25Hz

Table 2.5: List of *Dependent Variables* and their type and frequency.

Extraneous Variables	**Measurement**
Noise by other researchers in the lab	Put up signs to show that experiment is in progress. Kindly ask to be quite.
Temperature	Have a fan ready to ventilate the experimental space.
Fatigue	Short breaks in-between trials on a regular basis. If fatigue occured due to bad air/temperature, increase ventilation.
Early termination of the experiment by subject or experimenter due to unforseen reasons	Termination of the experiment. Save data as of last valid trial
Inappropriate fitting of prescription glasses	If prescription glasses are needed, perform eye test with provided eye test chart by Tobii.

Table 2.6: List of *Extraneous Variables*.

Confounding Variables
Subject does not appear
Power outage
Hardware failure

Table 2.7: List of *Confounding Variables*.

Chapter 3

Experimental Results

The experiment conducted in this Chapter received ethics approval from the Office of Research Ethics at York University (Certificates #2020-137 and #2020-217). Each subject signed a consent form as part of the ethics approvals. Appendix A provides a copy of the consent form.

This Chapter is an extension of a previous publication:

Markus D. Solbach and John K. Tsotsos "Active Observer Visual Problem-Solving Methods are Dynamically Hypothesized, Deployed and Tested", in *Presented at The Ninth Advances in Cognitive Systems (ACS) Conference 2021 (arXiv:2201.06134)*, [2021]

3.1 Introduction

In the previous chapters, we have presented the three-dimensional same-different task (Chapter 2), a complex real-world visuospatial task, and *PESAO* (Section 2.3), to record and analyze human subjects performing this task.

This chapter examines human visual behaviours for the three-dimensional same-different task based on a study with 47 participants completing in total 846 trials. The primary goal was one of discovery: We sought to explore and identify the characteristics of human behaviour in active tasks that could then inform the development of systems, such as robotic assistants that perform similar tasks. Figure 3.1 illustrates a sequence of fixations of a subject performing the three-dimensional same-different task. Left to right: sequence of consecutive fixations. Third-person view of subject within the *PESAO* facility (top row). Corresponding eye fixation (bottom row).

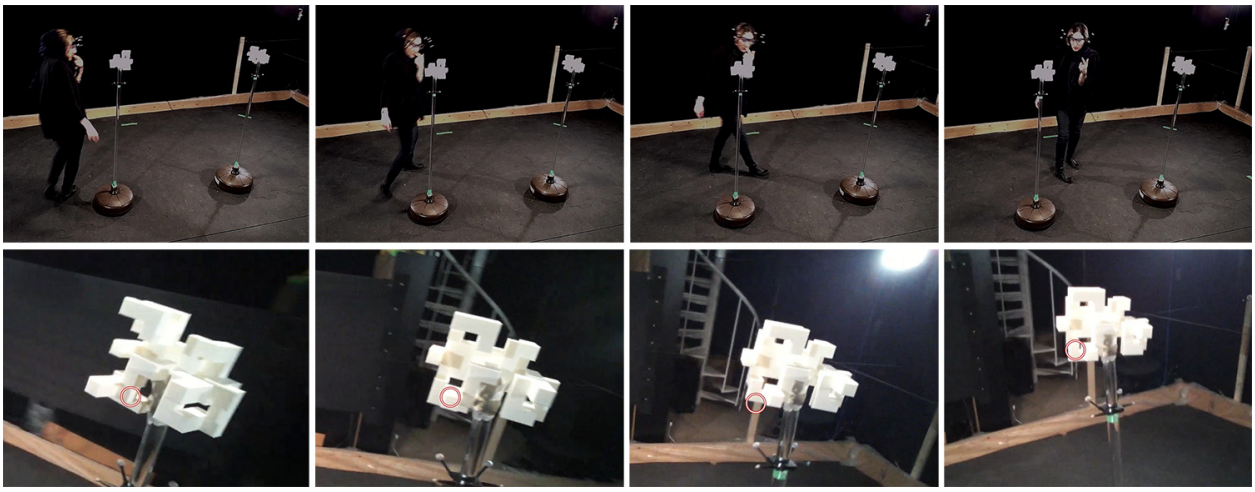


Figure 3.1: A sequence of fixations of a subject performing the three-dimensional same-different task. Left to right: sequence of consecutive fixations. Third-person view of subject within the *PESAO* facility (top row). Corresponding eye fixation (bottom row).

Our approach is different from what is proposed by [de Melo et al., 2021] where it is described that the next generation of computational learning systems will thrive through a shift from third-person data to first-person. Examples provided are [Grauman et al., 2022, Damen et al., 2018, Sigurdsson et al., 2018] and cover activities in different domains, such as laundry, folding, cleaning the dishes, preparing food, and so on. However, only a small portion of the Ego4D dataset by [Grauman et al., 2022] provides gaze information, and no dataset mentioned by [de Melo et al.,

2021] contains head tracking. To solve visuospatial problems, both elements, gaze and head motion[1], are crucial, hence describing the difference in our approach.

Following this section, the remainder is split into five sections. Section 3.2 provides necessary background for this chapter, details on the conducted experiment is given in Section 3.3, experimental results, such as amount of movement, number of fixations, et cetera are presented Section 3.4, and lastly, Section 3.5 summarizes this chapter.

3.2 Background

The study of human visual behaviour as an active observer has been limited. The comprehensive description of human visuospatial abilities in Chapter 8 of [Carroll, 1993] proved to be very helpful. We chose as a first exemplar the same-different task, a task humans need to solve often, and which seems an essential component of many other tasks.

Deciding if two objects are the same or different may seem straightforward. Often, we design objects to be easily discriminable, say by colour or size or pattern, but this is not always the case. Consider a task where you are given a part during an assembly task and need to go to a bin of parts in order to find another one of the same (for instance, see Figure 3.2 for assembly of furniture). Playing with interlocking toy blocks requires one to perform such tasks many times while constructing a block configuration, either copying from a plan, mimicking an existing one or building from one's imagination. There are many more examples. Obvious instances of this problem are not effective as probes into human solutions because humans are remarkable in their ability to home in on a workable strategy that can be used for most instances. We thus needed to push an experimental design to the extreme in order to discover the characteristics and limitations of the human solution space. The key question remains: What is the sequence of visual actions to correctly determine if two objects are the same? This problem has equal interest for human behaviour as well as robot behaviour.

In the current AI and computer vision community, one's first approach might be to learn solutions. It is quite likely possible to learn a viewing policy that simply covers all parts of a viewing sphere around each object, and then compares the feature representations on the sphere

[1]Not only pan and tilt as might occur while viewing a screen, but also change in position and orientation of the head.

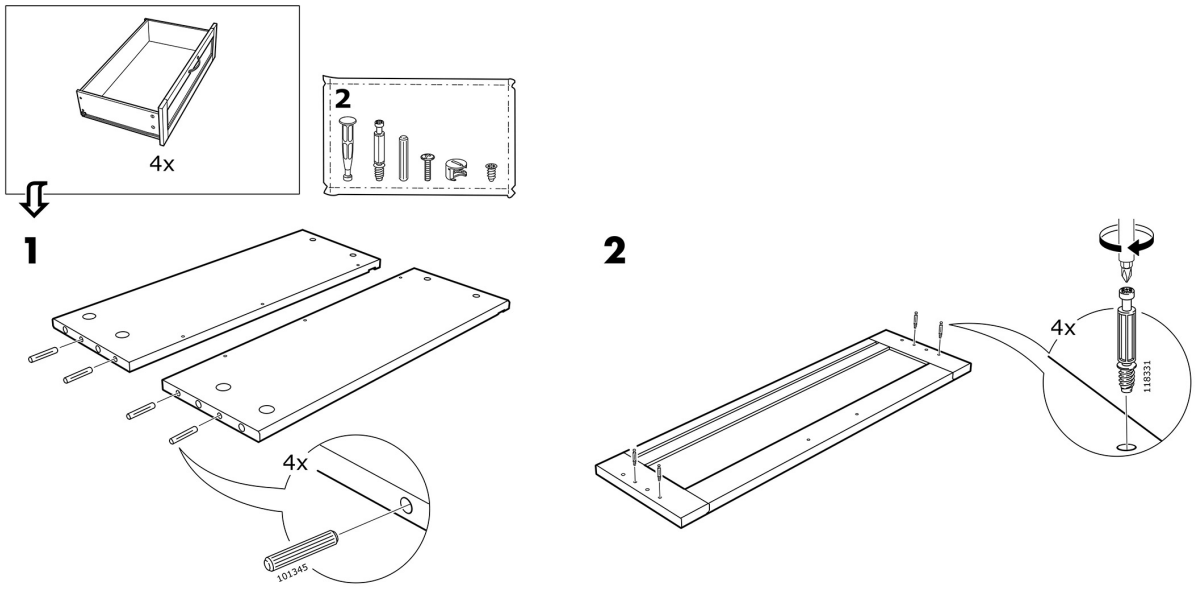


Figure 3.2: Example assembly instructions of IKEA furniture. These are the first two steps in the assembly of an IKEA drawer. They require at least twelve sameness comparisons to find all necessary parts and the tool. Often, furniture is assembled with different but very similar-looking screws, dowels and other parts, which makes it cumbersome to find the right one for a given step. Courtesy of Inter IKEA Systems B.V., accessed 15 March, 2022, Source: `https://www.ikea.com/ca/en/assembly_instructions/songesand-4-drawer-chest-white__AA-2021138-3.pdf`

surface. There are an increasing number of works that attempt to actively learn visuospatial tasks, incorporating prior knowledge to varying degrees and using multiple observations (for instance, [Settles, 2009, Ren et al., 2022]). But obvious, brute force solutions do not illuminate how humans do this in a far more efficient manner – we solve simple cases quickly, take an increasing number of views with increasing task difficulty, rarely need to see a full spherical view, and almost never take a single complete set of views. However, such conclusions seem subjective, and no experimental evidence is available that explicates how humans solve such tasks. The cognitive neuroscience community has also studied related visuospatial tasks, and as shown in [Tsotsos et al., 2021], they seem to be converging on the importance of flexible and dynamic composition of processing elements to achieve solutions for a given task. The experiment described in this chapter extends and suports this convergence.

3.3 Experiment

For the objects in this experiment, we choose the set shown in Figure 2.17. This set offers three complexity classes with four different objects each. Our definition of same and different is entirely based on the geometrical configuration of the objects. Same means that both objects consist of the exact same configuration of elements (including size), whereas different means that at least one element of the geometrical composition of at least one object is different. In order to analyze the active behaviours of observers, we track the head motion and the participants' gaze and allow natural movements with subjects untethered and without predetermined viewpoints.

All variables (Section 2.4), except object complexity, are chosen randomly using a uniform distribution for each trial. For object complexity, we opted for a pre-defined set of six trials of each complexity level; however, the order of presentation was randomized using a uniform distribution. No patterns due to biases in the sampling were observed.

To date, we have conducted 846 trials of the three-dimensional same-different task with 47 participants. Two objects are selected and mounted on posts in the experimental space. There are multiple independent variables to explore: objects may be same or different, object complexity may be easy, medium or hard (as shown Figure 2.17), the object pose or mounting orientation difference between their base plates may be 0.0°, 90.0° or 180.0°, and the subject's starting position may be at the long (close to targets, each equidistant from the subject), corner (subject oblique to targets, one closer than the other), or short positions (targets are in line and subject views along that line, one object farther than the other).

All are chosen randomly for each trial, and subjects are placed into the experimental space facing away from the objects at one of the three starting positions. Once the subject is instructed to turn towards the objects, the trial begins, and all head and gaze movements are recorded, as are the first-person and third-person videos of the entire trial. The first fixation that counts is the one that falls on a target object. After the trial, subjects complete a questionnaire that asks about strategies and observation methods. Figure 3.3 shows a sample gaze trace for a particular trial.

There was no hypothesis that these experiments were testing, and there was no learning task for the subjects. As mentioned, this was an experiment of discovery. We sought to explore and identify the characteristics of human behaviour in active tasks because little seems known about

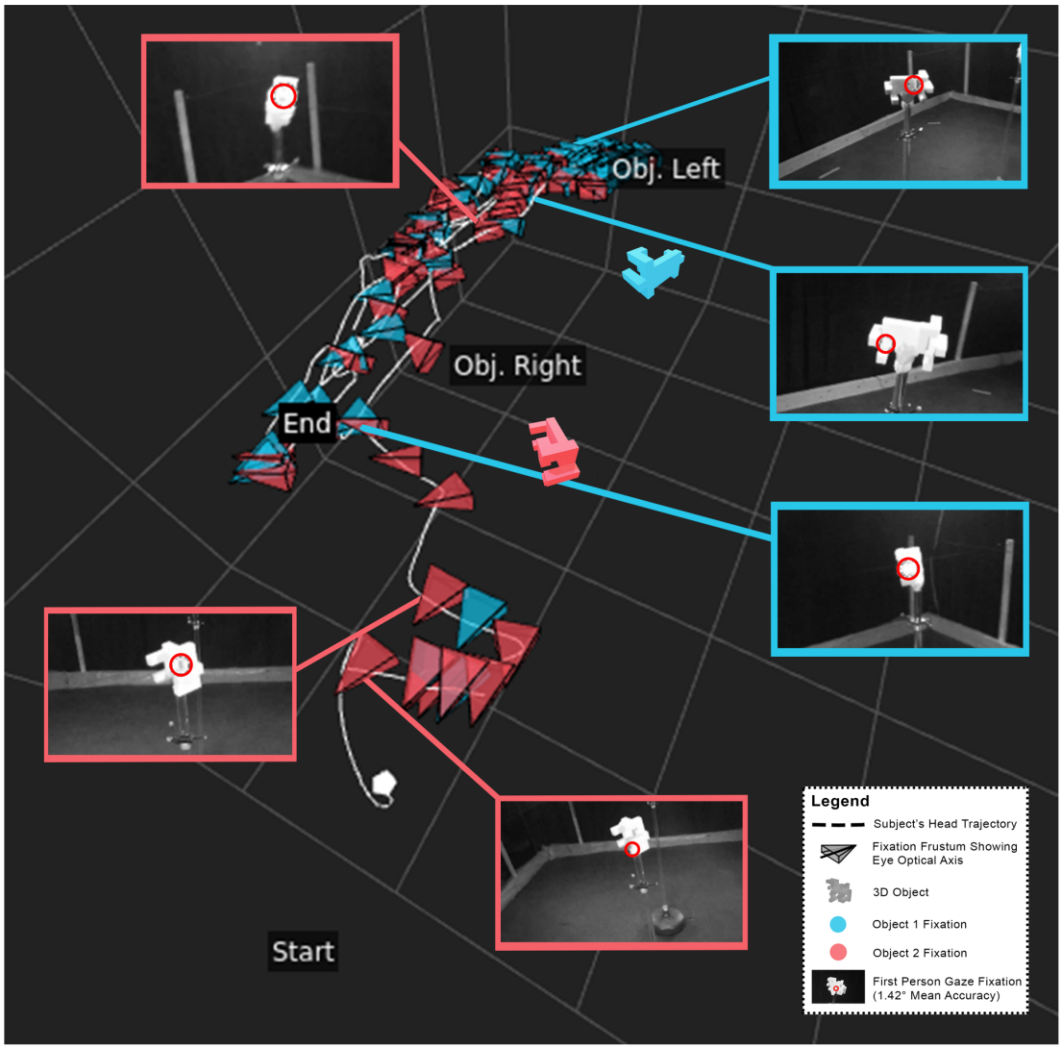


Figure 3.3: A visualization of the recorded data using PESAO. The subject's movement is plotted as a dashed line in white, and fixations of either object are illustrated as a fixation frustum in the corresponding colour of the fixated object. Selected fixation frusta are annotated with snapshots of the subject's first-person view and the gaze at a particular fixation (red circle). In this example, the objects are the same, they differ in the pose by 90°, and the subject started from the short position.

this. What we find could then inform the development of artificial systems that perform similar tasks.

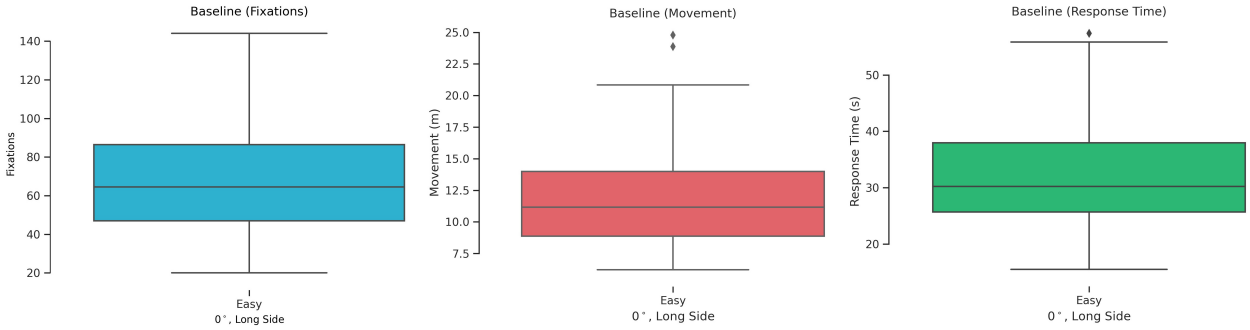
3.4 Results

We considered several different performance metrics such as accuracy, response time, the number of fixations and more. All of these were computed with respect to the different variables used

in this study: starting position, object complexity, orientational difference, and target sameness. Furthermore, we also investigated how a correct or incorrect trial affects these measures.

3.4.1 Baseline

Figure 3.4 presents the baseline evaluation of this experiment[2]. It shows the experimental set up with the highest accuracy for the number of fixations (Figure 3.4a), amount of movement (Figure 3.4b), and response time (Figure 3.4c). The experimental set up that achieved the highest accuracy, namely 100% in all trials, had the set up of object complexity of "Easy", object orientation of 0°, and starting from the long side.



(a) **(b)** **(c)**

Figure 3.4: Plots illustrating the baseline for the most accurate experiment combination (Complexity, Orientation and Starting Position) with respect to the number of fixations **(a)**, amount of head movement **(b)** and response time **(c)**.

The baseline case required on average about 65 fixations, which is about a third lower than the absolute mean of the experiment (Figure 3.7). An example of the baseline case is shown in Figure 3.5. It illustrates an example initial view assuming a subject height of 1.6m[3].

The trial with the least fixations needed 20 fixations, and the one with the most, 140 fixations. One can see that there is large variations in how the baseline case is approached in terms of the number of fixations. In terms of head movement, the baseline average is at 11m and ranges from about 5 to 25m. The response time is on average 30 seconds, with absolute values going from 10-70 seconds.

[2]Note: These and all following box plots show the three quartile values of the distribution along with extreme values. The whiskers extend to points within the 1.5 interquartile range of the lower and upper quartile, and observations that fall outside this range are displayed independently as diamonds.

[3]The objects are "different".

Figure 3.5: Example initial view of the baseline set up. The experimental set up of the baseline consists of objects of easy complexity, an orientation of 0°, and starting from the long side. This figure illustrates what the initial view might look like if the subject looks at both objects perfectly straight ahead.

This hints at a larger variation of how this case is addressed by the subjects – deploying different strategies that require different amounts of fixations, movements and response times to be executed.

3.4.2 Accuracy Data

Humans are remarkably good at this task (Figure 3.6a). Throughout all evaluated combinations of this task, participants achieved an absolute mean accuracy of 93.82%, $\sigma = 3.9\%$. The best performance was achieved for the complexity level easy, with subjects starting by viewing from the long position with an orientation difference of 0.0°. These objects are the least complex, making them the easiest to compare.

Secondly, starting from the long position presents both objects at the same distance and next to each other immediately from the start. An orientation difference of 0.0° means that the objects are also aligned in their orientation, simplifying the comparison. Further, for all complexity classes, trials starting from the long position always resulted in the best performance within a complexity class: 96.96% (easy), 94.52% (medium), and 96.2% (hard). By contrast, objects of complexity

hard for trials that started from the short position have the worst mean performance of 90.38%. However, overall, no significant effect of the starting position on the accuracy is shown ($F_{2,92} = 2.11, p = 0.125$, where the subscript of F describes the degrees of freedom for each variable. Here, *starting position* and *accuracy*)[4].

We compared accuracy and fixation numbers for individual subjects across the number of trials each subject performed (see Figures 3.6d and 3.7d). We have 47 subjects, each subject did 18 trials (6 for each complexity level), and no target object configurations were repeated. Nevertheless, we expected that some improvement in performance would begin to appear.

This was not the case; there is no significant learning effect in accuracy ($F_{5,230} = 0.836, p = 0.525$). No learning effect can be seen in the case of easy complexity. A minimal effect can be seen for the complexity classes of medium and hard. However, none are significant ($F_{2,92} = 0.888, p = 0.414$).

We also evaluated the effect of the relative orientation of the objects (see Figure 3.6b). For the easy case, there is a clear gradient of accuracy following the increase of orientation difference. Notably, trials of easy with orientation 0.0° had an accuracy of 100%. However, for the other complexity levels, a different pattern can be identified; 90° was most accurately identified with 94.82% and 90% for medium and hard, respectively. 0.0° and 180° ranked second and third. Orientation difference also seems to impact the number of gaze shifts and head movements required. The amount of head movement and response time increased with orientation difference at all complexity levels, but a statistical analysis showed that the effect of object orientation on the accuracy is not significant ($F_{2,92} = 2.06, p = 0.132$).

Lastly, a significant effect of object "sameness" on the accuracy exists ($F_{1,46} = 3.58, p = 0.044$). Figure 3.6c shows a remarkable point: the deviation of accuracy is much larger for different cases than for same. For instance, objects of complexity easy have been accurately identified as "same" within the upper and lower quartiles ranging from 94% to 100%, whereas "different" cases ranged from 90% to 100%. This effect is very dominant for the hard cases; "same" ranges from 92.5% to 94.5%, but "different" starts at 87.5% to 100%. This is, however, less dominant for the medium

[4]Our statistical significance analysis is performed using one-way repeated-measures ANOVA utilizing the software library provided by [Vallat, 2018]. Other statistical analysis, such as two-way repeated-measures ANOVA, might give further insight into our data but is, at this point, left for future work. For instance, our analysis looks at one independent variable's effect on a dependent variable. However, our subjects were exposed to multiple independent variables at the same time.

cases (94-100% vs. 86-93%). Furthermore, except for easy cases, the median accuracy for medium and hard objects is higher for the same object pairings. For the easy case, the median is about 2% lower (96 vs. 93.82), but the upper and lower quartiles are narrower.

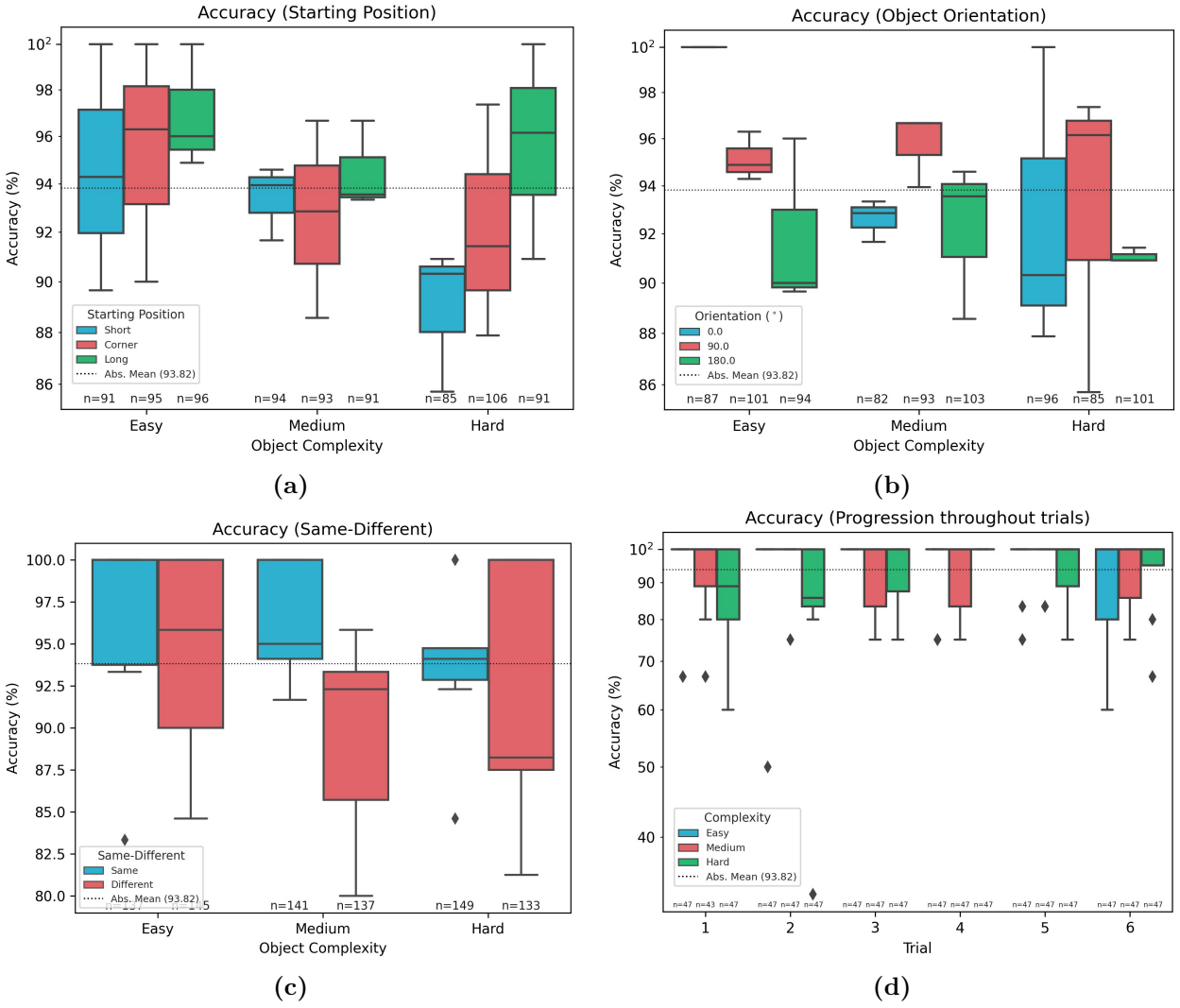


Figure 3.6: Results illustrating the accuracy measured against different experimental variables: Starting Position **(a)**, Object Orientation **(b)**, Sameness **(c)**, Progression throughout trials **(d)**.

3.4.3 Fixation Data

Across all trials, the absolute mean number of fixations was 92.38 (see Figure 3.7a). No trial took fewer than 6 fixations. Trials with just 6 fixations can be identified as trials starting from the short side, with an object orientation difference of 180°, object complexity of medium, objects being different, and provided a correct answer (See Figure 3.7c, 3.7b and 3.7e).

There is an important point here. Even our simplest case required six fixations, meaning six feedforward passes through the visual system, with possible top-down components for each, in addition to the computation and execution of eye fixation itself. Models of vision that employ a single feedforward processing pass with no eye movements seem unsuited to this task or to comparison with human vision.

Figure 3.7b presents the number of fixations against the different starting positions ($F_{2,92} = 1.37, p = 0.258$). The long starting position seemed to require the fewest fixations, the short position required the most for medium and hard objects, while the corner position required the most for easy objects. Overall, the complexity of objects significantly influences the number of fixations needed ($F_{2,92} = 32.15, p < 0.0001$).

Figure 3.7b shows that greater orientation differences require more fixations for all three object complexity levels ($F_{2,92} = 8.31, p = 0.00048$). However, orientations 0° and 90° are similar, varying only a few fixations for the median and upper and lower quartile. In terms of absolute values, a few trials of complexity hard and orientation of 0° required about 800 fixations. Notably, these trials (when compared to Figure 3.7a) started from the long side. This shows that even though this is a configuration of higher accuracy, it does not necessarily mean it requires less observation to be answered.

This observation also holds for the following; same responses, while generally higher in accuracy, require roughly 10 to 20 fixations more (see Figure 3.7c).

Looking at the progression throughout trials to see if there is a learning effect, which in this case would mean that the number of fixations decreases, we have plotted the number of fixations against all six trials of easy, medium and hard each in Figure 3.7d. Like the analysis of the accuracy (Section 3.4.2), a learning effect is not noticeable ($F_{5,230} = 3.239, p = 0.0075$). This also means that the task is solved more efficiently as the trials were executed.

Figure 3.7c illustrates the number of fixations with respect to the sameness of objects. For all instances, the "same" pairing of objects required significantly more fixations ($F_{1,46} = 7.78, p = 0.0076$) than "different" objects.

Lastly, a significant effect is shown for the correctness of the answer and the number of fixations (Figure 3.7). Generally, correct answers needed fewer fixations than false ones ($F_{1,46} = 9.762, p = 0.0030$). One might think that this is due to uncertainty when providing an incorrect answer.

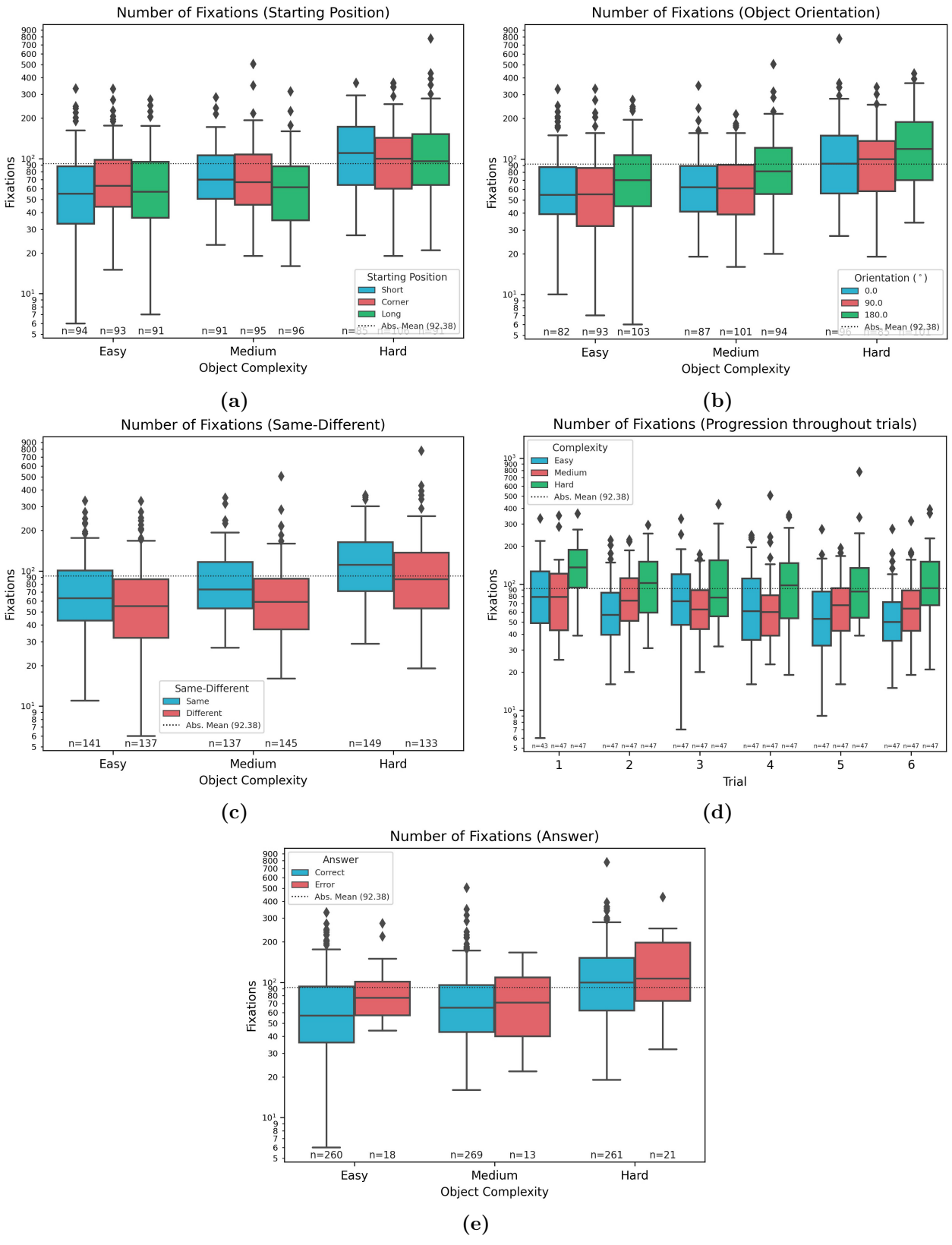


Figure 3.7: The number of fixations against different experimental variables: Starting Position **(a)**, Object Orientation **(b)**, Sameness **(c)**, Progression throughout trials **(d)**, Correct/Error Answer **(e)**.

3.4.4 Response Time Data

Response time is the time elapsed from the start of the trial (the subject turns around and faces the objects) to the end of the trial. On average, the response time over all trials was 47.52s, with σ=30.39 from start to answer (see Figure 3.8a). Among all our trials, the shortest response time is for easy cases starting from the short position (4.2s), whereas the longest response time is 298s for a hard case from the long position.

In terms of object orientation difference, the lesser orientation difference also means a quicker response time ($F_{2,92} = 12.95, p < 0.0001$). For all object complexity levels, 0° was the fastest to be answered, followed by 90°, and 180°. Generally speaking, increasing object complexity also means a significant increase in response time ($F_{2,92} = 28.87, p < 0.0001$). While easy and medium cases are mostly on par, an increase in response time can be seen for the hard object case.

Object pairings of "same", as we have discussed before, required more fixations than "different". This trend is also noticeable for the response time ($F_{1,46} = 14.279, p = 0.0004$). "Same" cases take longer than "different" ones which also means that a fixation, even though more are taken, still takes a similar amount of time to be executed. Furthermore, "different" cases also hold the record for the fastest and slowest response. The fastest response was 4.5s for a trial of easy complexity, starting from the short side and with an orientation difference of 180°. On the contrary, the slowest response was 300s (5 Minutes) for a trial of hard complexity, starting from the long side, with an orientation difference of 180°. Overall, the starting position (Figure 3.8a) did not have a significant effect on the response time ($F_{2,92} = 0.12, p = 0.886$).

While no learning effect was identified analyzing the accuracy, a slight decrease in response time over the trials can be seen for the medium complexity case. Figure 3.8d illustrates how the response time developed through all six trials of each complexity level.

Starting at about 47 seconds (median) at the first trial, the response time drops to about 34 seconds (median) for trials two to four and drops further to 29 seconds median at trials five and six. For hard cases, a drop from the first trial (70 seconds median) to the second trial (about 50 seconds median) can be seen. However, the following trials stay at about the same. Lastly, easy complexity cases vary across the trials with no noticeable trend. A possible explanation is that perhaps these response time observations mean that the easy cases are easy enough that no learning was required,

that the hard cases are too hard to be learned over just six trials and on the contrary the medium cases allowed for some response time improvement to happen, namely two times; one after the first trial and the other after the fourth trial. Overall, just looking at the impact of progressing trials and their response time, a significant effect is noticed ($F_{5,230} = 6.01, p = 0.0003$).

While incorrect answers are few, it can be noted that the response time, in general, is significantly higher than for a correct answer ($F_{1,46} = 18.62, p < 0.0001$). Figure 3.8e shows the results. While the difference for the medium case is about 20s, the difference is less prominent for the easy case with around 9s and roughly 11s for the hard case.

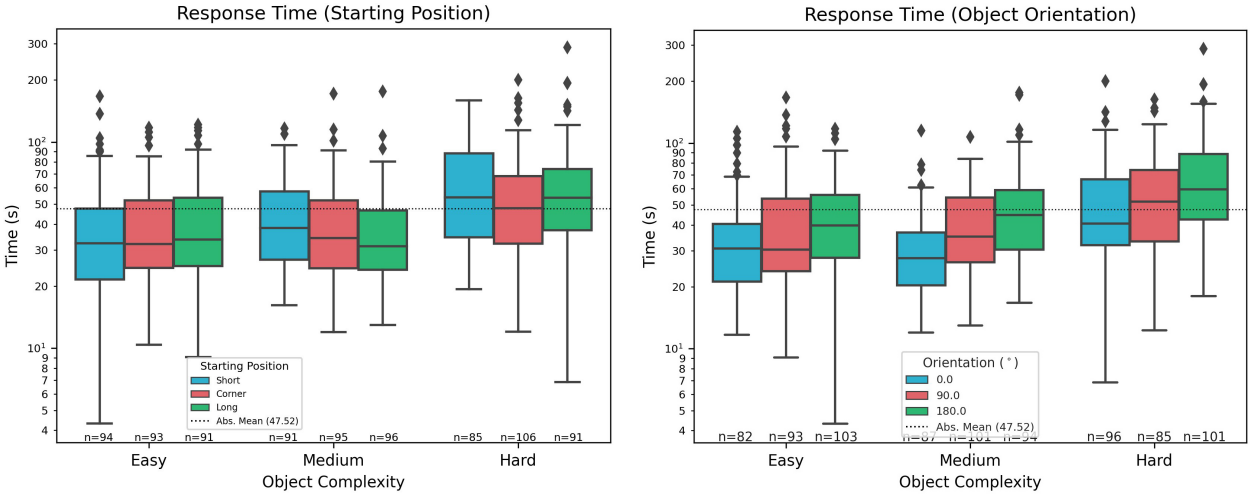
This shows that uncertainty about the answer prolongs the response time and results in more extended observations. While we can see a drop in response time of about 0.26 seconds from complexity level easy to medium (40.79s and 41.05s, respectively), the response time increases to 60.61s for the hard cases. We conclude that though the objects are roughly of the same size, the increase in visual features demands more time.

3.4.5 Head Movement Data

A key metric to evaluate for this experiment is the amount of head movement required to solve this task. This section will be going through five different combinations to understand how and what affected the amount of head movement displayed. Figure 3.9 presents all five plots, and we will be going through them one by one.

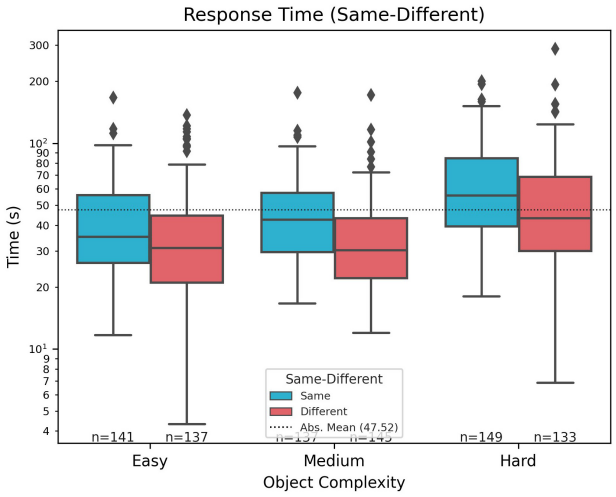
The absolute mean of head movement was 16.62m over a trial, and no trial had less than 1m of head movement (see Figure 3.9a). 1m of head movement is a substantial displacement to vary the position of the sensory apparatus. The objects are 1.2m apart from each other, and considering the most accurate starting position (long side), the subject starts 1.7m away from the objects. This said the subject is able to observe both objects without moving the head at all (but moving the gaze). Yet, deploying the sensory apparatus was always observed.

The amount of head movement slightly increased from complexity cases of easy to medium but increased more distinctly for hard cases – the object complexity significantly affects the amount of head movement ($F_{2,92} = 35.35, p < 0.0001$). A clear trend ($F_{2,92} = 22.74, p < 0.0001$) can be observed between the amount of head movement and amount of orientational difference (see Figure 3.9b); if both objects were aligned (0°) in all complexity cases, the least amount of movement was

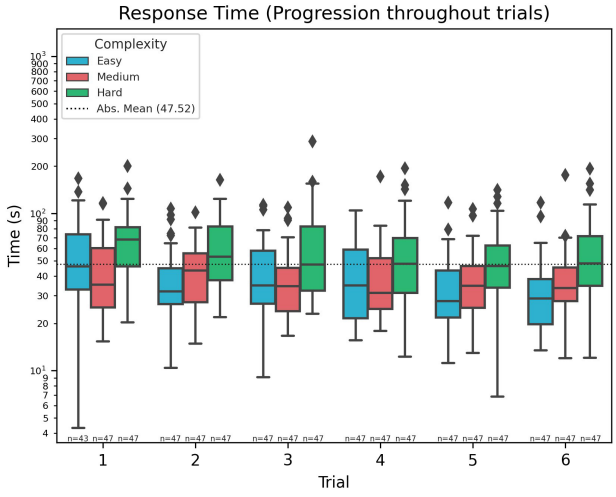


(a)

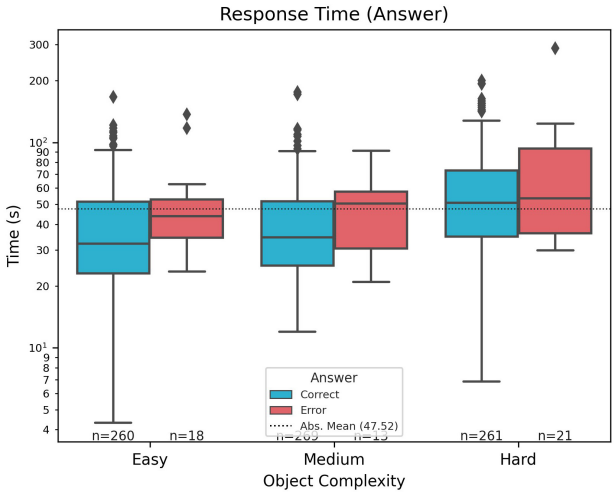
(b)



(c)



(d)



(e)

Figure 3.8: Response Time against different experimental variables; Starting position **(a)**, Object Orientation **(b)**, Sameness **(c)**, Progression throughout trials **(d)**, Correct/Error Answer **(e)**.

required, for 90° an increase of 2-5m on average was recorded, and finally, for the 180° an additional increase of 1-5m was recorded. This confirms the findings of [Shepard and Metzler, 1971] but in three-dimensions and allowing for active observation[5].

Aligned with the number of fixations, response time and accuracy, the amount of movement necessary is more for the same object pairings across all complexity levels. A significance analysis also confirms this as well: $F_{1,46} = 31.37, p < 0.0001$. Figure 3.9c illustrates the data. Generally, also the upper and lower quartiles are narrower for this case, showing that less variation in the amount of head movement is present. For different cases, the increased upper and lower quartiles indicate that more uncertainty across different subjects in how to approach this case was involved.

To analyze the effect of learning, we have plotted the amount of head movement across all six trials of each complexity class in Figure 3.9d. Learning here would mean that the amount of head movement would decrease as deployed strategies become more efficient and require less movement. We see no evidence for this effect for the medium and hard object complexity cases but some for the easy cases. Hard cases start of at the first trial with just above 20m and drop to the absolute mean value of 16.62m and stay there, marginally falling below and exceeding it repetitively. Similarly, for the medium case, where no learning trend can be observed. However, the easy case, while noticing a slight up-trend for the second trials, consecutively lowers from about 16m, down to about 10m, which is an improvement of 37.5%. A significant analysis, however, reveals a significant reduction in head movement over the trials ($F_{5,230} = 5.403, p = 0.0001$) – just like with the number of fixations and response time, the strategies seem to get more efficient over time.

Lastly, error responses (see Figure 3.9e) were accompanied by significantly more head movement ($F_{1,46} = 41.56, p < 0.0001$). As stated before already, we have over ten times more correct than error responses. So, it is to no surprise that the upper and lower quartiles of error responses are narrower compared to correct responses. However, the trial that required most head movement was an error response, with complexity hard, recorded during the third trial, with different objects, starting at the long side, with an orientation of 180°. This trial required almost 100m of head movement, which is the same as going back and forth between the objects physically about 83 times.

[5] "The time required that two perspective drawings portray objects of the same three-dimensional shape is found to be a linearly increasing function of the angular difference in the portrayed orientations fo the two objects." [Shepard and Metzler, 1971]

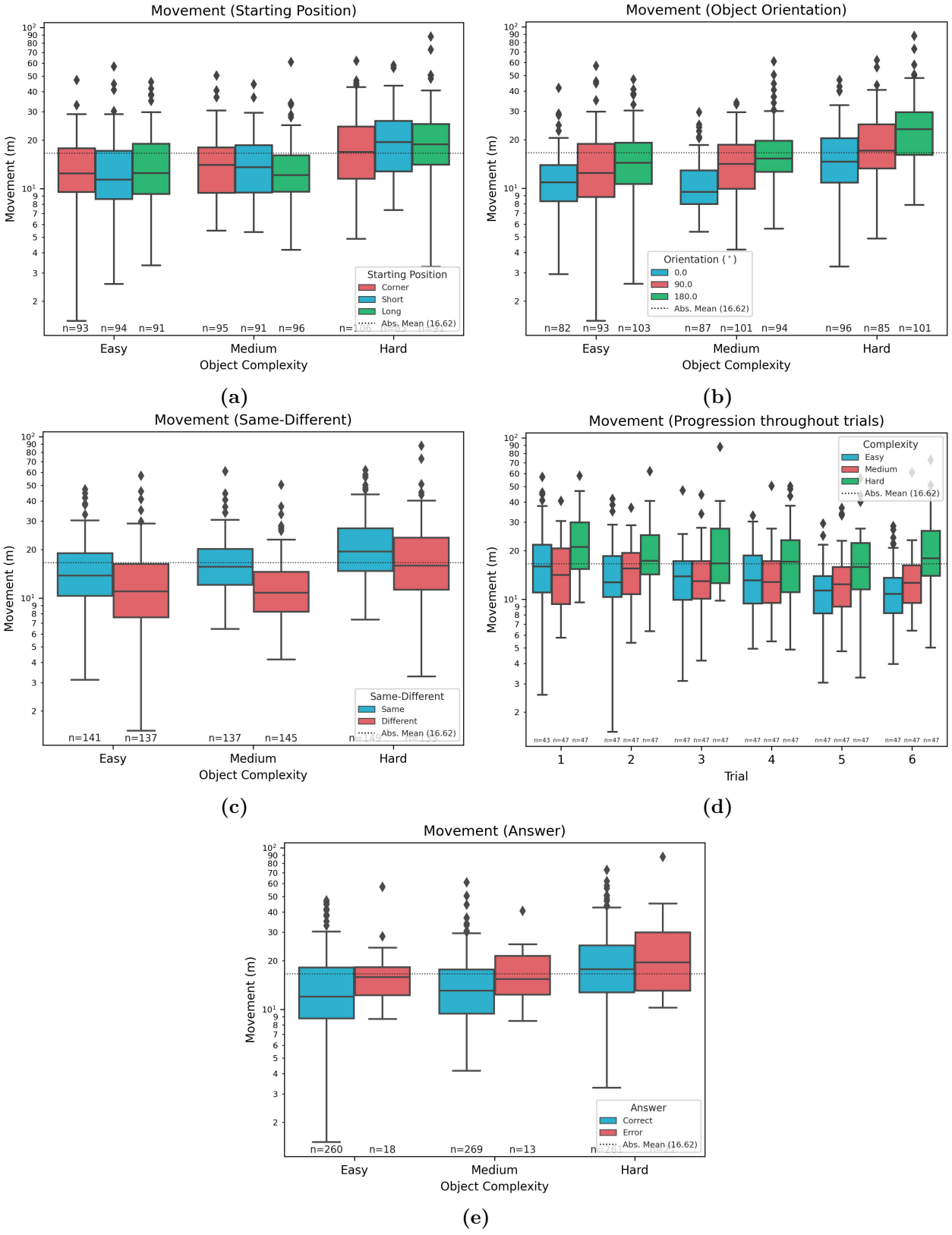


Figure 3.9: Head Movement against different experimental variables; Starting position **(a)**, Object Orientation **(b)**, Sameness **(c)**, Progression throughout trials **(d)**, Correct/Error Answer **(e)**.

3.4.6 Comparison to Optimal Algorithms

In this section, we turn towards computational solutions for solving the same-different task. We are not interested to see which algorithms would solve this task – they already exist – we want to know if humans solve this task differently.

One approach to solving this task computationally is to check for congruence and symmetry of the two three-dimensional objects. This has been studied in a number of papers and is considered optimally solved in $\mathcal{O}(n \log n)$, where n describes the size of the geometrical shape [Sugihara, 1984, Atkinson, 1987, Alt et al., 1988, Jiang and Bunke, 1991, Jiang et al., 1996, Braß and Knauer, 2002]. These are only for finite point sets or convex polytopes and do not generalize to other sets of objects. [Brass and Knauer, 2004] proposes a computational algorithm for congruence and symmetry check of general three-dimensional objects and proofs that this can be done in $\mathcal{O}(n \log n)$ time as well.

$\mathcal{O}$ maps the input size n to the run time t; for this, we need to specify what n is for the three-dimensional same-different task. It is intuitive to assume that the complexity level of an object plays a crucial role in this. [Brass and Knauer, 2004] tests congruence and symmetry for general three-dimensional objects. They define the object complexity as $n = \#O$ where O is the number of subobjects[6]. Subobjects are of constant description complexity, and therefore computations of congruence can be done in constant time.

Similarly to [Brass and Knauer, 2004], for the three-dimensional same-different task, as proposed here, the subobjects are the number of building blocks (cuboids and base) used to create the object (Figure 2.15). So, the components number for objects of easy is $n = 7$, medium is $n = 10$, and hard is $n = 18$. The response time (for instance, Figure 3.8c) for each of these are easy $= 40.79s$, medium $= 41.05s$, and hard $= 60.61s$.

In order to visually understand the complexity class for our data, in Figure 3.10, we plot them against common complexity levels, as well as the complexity level of the algorithm proposed by [Brass and Knauer, 2004].

We can see that the algorithm used, on average among all subjects, lies below $\mathcal{O}(n^2)$ and above $\mathcal{O}(n \log n)$. Nonetheless, $\mathcal{O}(n \log n)$ increases faster for larger n than $\mathcal{O}(h)$ and will likely surpase

[6] $O \neq \mathcal{O}$. $\mathcal{O}$ is the symbol used for the Big O notation.

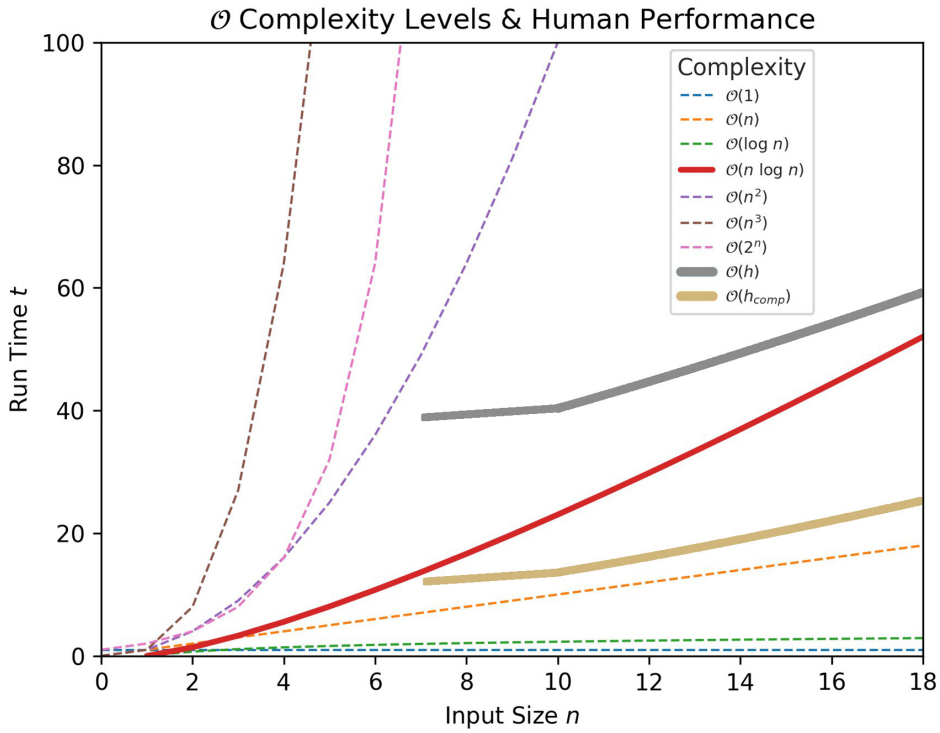


Figure 3.10: Illustration of a few Big $\mathcal{O}$ complexity classes including provably optimal algorithms descibed in the text $\mathcal{O}(n \log n)$ in red, the human strategy complexity $\mathcal{O}(h)$ for the response time in grey and the compensated response time $\mathcal{O}(h_{comp})$ only accounting for the time spend fixating on one of the objects in gold.

it for $n > 20$.

However, a crucial advantage of computational approaches is that three-dimensional information can be acquired virtually for free and is immediately available, which is not the case for humans – humans need to move their heads, change their viewpoint, and so on. Perhaps a better measure than looking at the response time is to look at the fixations; specifically, the amount of time spent performing fixations. It must be noted that the time between fixations is also used, for example, reasoning and planning.

The complexity for this is shown in Figure 3.10 as $\mathcal{O}(h_{comp})$. Taken from our data, the average fixation time is 200ms. We use the average number of fixations for easy = 76.56, medium = 79.29, and hard = 121.06 (for instance, Figure 3.7c). This results in a compensated fixation time of easy = 15.31, medium = 15.8, and hard = 24.21. This sets the human performance between $\mathcal{O}(n \log n)$ and $\mathcal{O}(n)$. In comparison to h the slope for h_{comp} cannot be interpreted to be less than the slope of $\mathcal{O}(n)$. However, it is clearly below $\mathcal{O}(n \log n)$, hence less complex than the computational algorithm

presented by [Brass and Knauer, 2004].

Another observation became clear in this analysis when looking at the slope of h and h_{comp}. Between medium and hard, the slope of h is steeper than the one of h_{comp}. This indicates that less time is spent between fixations and, therefore, less processing between fixations is done – not only a change of strategy but also a change to a strategy that requires more data, perhaps one of the processing elements in more detail.

In conclusion, while only three data points are available at this point (easy, medium and hard), the run time with respect to the input size n to solve the three-dimensional same-different task seems to be describable as $\mathcal{O}(n)$. However, more experiments with different input size n are needed to draw a more conclusive answer. Furthermore, taking into account that our subjects had not seen the objects before and were not particularly prepared to solve the same-different task per se, it is remarkable to observe a complexity of about $\mathcal{O}(n)$ which is lower than an optimal algorithm; hence humans solve this task differently.

3.5 Summary

This chapter outlines our steps towards an understanding of how agents (human or artificial) might solve complex three-dimensional visuospatial tasks as active observers. We discovered that there was no human experimental data available to inform our work.

- People are very good at this task even for difficult cases. The range of response times from simplest to most difficult cases ranged from 4 - 298 sec. and accuracy from 80% to 100%.
- It seems that humans solve the *Three-Dimensional Same-Different Task for Active Observers* efficiently with a run time complexity of about $\mathcal{O}(n)$ where n describes the size of the geometrical shape as proposed by [Brass and Knauer, 2004].
- There is a great deal of data acquisition occurring during all trials with the range of eye movements (and thus separate fixations and separate images processed) from 6 to 800 fixations.
- No statistical change has been observed in accuracy with increasing trials for individual subjects. However, change has been observed for the number of fixations, response time, and overall head movement.

In the next chapter, we analyze our data and present strategies and methods commonly found when subjects approached the *Three-Dimensional Same-Different Task for Active Observers*.

Chapter 4

Analysis

The work in this Chapter is an extension of a previous publication:

Markus D. Solbach and John K. Tsotsos "Active Observer Visual Problem-Solving Methods are Dynamically Hypothesized, Deployed and Tested", in *Presented at The Ninth Advances in Cognitive Systems (ACS) Conference 2021 (arXiv:2201.06134)*, [2021]

4.1 Introduction

As the data presented in the previous chapter imply, the goal of finding structure within sequences of hundreds of actions encompassing about 80,000 fixations is a daunting task. In this chapter, we turn to data mining techniques to discover common patterns and also examine the subject questionnaires to analyse the results.

Following this section, in Section 4.2 we introduce our method to mine and analyse fixation sequences, discuss implications for cognitive programs in Section 4.3, and conclude this chapter in Section 4.4 with a summary.

4.2 Mining the Fixation Sequences

Most data-mining approaches were not helpful; even the popular Trajectory Pattern Mining Algorithm [Giannotti et al., 2007] did not bear fruit finding either nothing or large meaningless patterns. We examined the subject questionnaires for clues. For example, a common report was that subjects look at one object then the other, searching for the presence of particular structures seen in one object on the other. This and other such reports provided a few abstract anchors, and using these we embarked on hand-labelling the data. We then looked through the first-person videos, annotated the gaze and looked for commonalities. When found, each was formalized and checked across other trials. Table 4.1 provides the result of this procedure, as well as three example answers from the questionnaires.

Each item was considered if more than ten subjects ($\approx 20\%$) reported it. The table also reports the occurrance for each item in the recorded data. This number is higher, as the subjects were only interviewed once after the experiment and perhaps forgot every detail about their approach. Further, items "Tracing Connected Components", "Comparing Arbitrary Components", "Point of View Change", and "Viewing Angle Change", which we will categorize as "Elemental Operations", were solely derived from the recorded data.

Based on the recorded data, we were able to create a flow-chart, which allows us to categorize the items and define a rough temporal order. Figure 4.1 shows the entire flow-chart of our findings which we will describe now in more detail.

Item	Occurrance: Questionnaires/Data	Reported Examples
3D Layout	12/35	• "I familiarized myself with the room" • "I checked from where I can look at the objects" • "I checked where I can walk"
Localization of Target Objects	19/45	• "I checked where the objects were" • "I checked from where I can look at the objects" • "I didn't know where the objects were. I had to look for them first"
Global Gist	27/47	• "I tried to get a rough idea of the objects" • "I walked around each object to check for the baseplate, which was easy to identify" • "First, I checked where the forward-facing elements is by quickly walking around the objects"
Outlier Detection	10/19	• "Sometimes it was very obvious as there was an element that didn't exist on the other object" • "I checked if that structure existed on the other as I didn't remember that it did" • "This element (pointing at a part of the object) doesn't exist here (pointing at the other object)"
Divide and Conquer/Coarse to Fine	32/47	• "For more complex objects I had to look at one part as a whole and then look at details of it" • "I checked a certain part of the object; when I saw it on the other object, I checked its elements" • "Sometimes, I was able to look at a big part of the object and compare it with the other object. When they still looked the same, I checked smaller details"
Alternating Fixation	18/47	• "From a certain point, I was able to compare the objects" • "Sometimes I could look back and forth to look at the objects without moving" • "When both object were facing in one direction, I could easily compare them without walking too much"
Alternating View	16/47	• "I found it hard when the objects were not facing the same direction. I had to move around to compare them" • "I used the structure of the objects to know where to walk. I preferred to look from a certain angle at the objects first. But sometimes, I need to walk to realize this" • "I compared parts of the objects. When I was not able to directly compare them, I used my hands to remember the shape, walked to the other object and tried to compare"
Strategy Repetition	15/21	• "Sometimes I had to double-check before answering" • "I wanted to answer but forgot if the element was going out left or right" • "I revisited the same locations to make sure"

Table 4.1: Reported strategies that occurred more than ten times in the entire experiment. Provided are three examples for each item and the occurrence found in the data for this item. We report the occurrance per subject (18 trials) and not per trial.

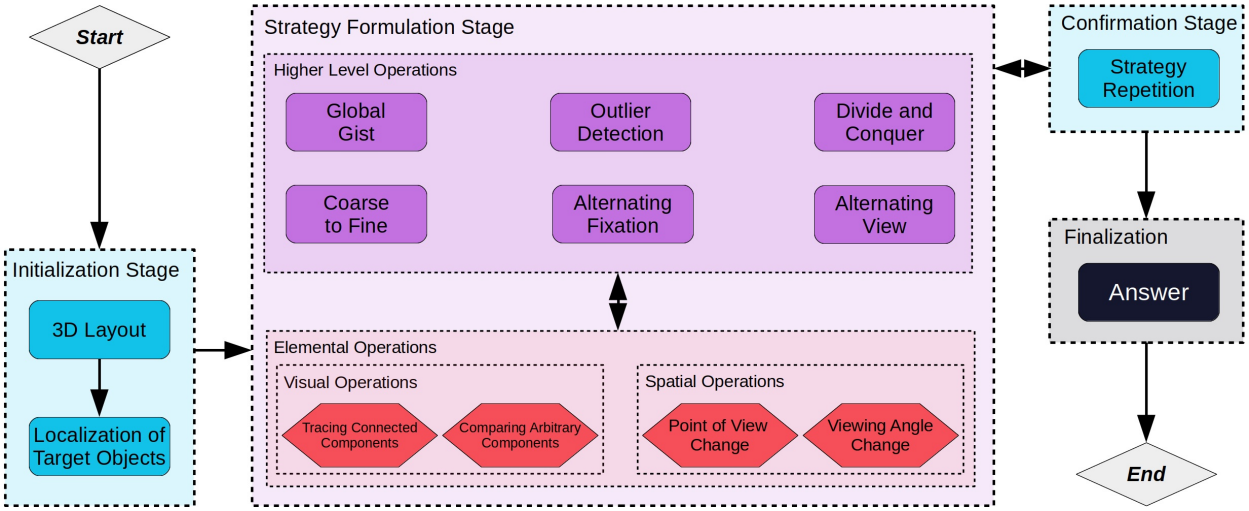


Figure 4.1: A diagram showing how visuospatial strategies are composed to solve the three-dimensional same-different task. Three different stages were identified: Initialization, Strategy Formulation, and Confirmation. Each stage contains different operations and elements.

A. The *Initialization* stage is usually a short time window between 3-5s in which the subject performs two routines to set themselves up for the development and deployment of a strategy.

   1. *3D Layout* – Subjects seem to first evaluate the overall 3D layout. After receiving the start signal, the subject turns away from the black curtain and assesses the scene for traversability and to self-localize within the scene. This is detectable as a quick scan of the environment, such as panning the field of view from one side of the experimental setup to the other.

   2. *Location of Target Objects* – The second routine deals with localizing the objects within the scene. The subject needs to know where the objects are and their spatial relationship in order to plan the first spatial operation. This operation is often intertwined with 3D Layout; distinct, short fixations ($< 300ms$) of both objects, either during or after 3D Layout.

B. Subjects seem to think about strategy. The *Strategy Formulation* stage is the core stage, which might be repeated several times by developing, deploying and dismissing different strategies (hypothesize-and-test). A strategy is comprised of one or more, with possible recurrence, of

   1. *Global Gist* – This describes how much of the virtual viewing sphere around each object has been explored. We quantize the viewing sphere into eight sectors. A sector is

considered visited if fixation is carried out from within it. While the head pose needs to be within this sector, the gaze data is used to determine whether the object is fixated.

2. *Outlier Detection* – An outlier is an element of one object that differentiates from the other object and is determined using only the gaze information. If the subject observes the differentiating part of the object which was previously observed on the other object and finalizes the trial with an answer, outlier detection is noted.

3. *Divide and Conquer* – This operation divides the object into smaller parts which are used for comparison. If the part has been divided and compared with the other object, this part is considered "conquered." To detect this operation, the head pose, as well as the gaze, is used to recognize which part is being sub-divided and compared. A conquered part is not further sub-divided but can be revisited.

4. *Coarse to Fine* – Similar to Divide and Conquer, this operation reduces the visual information, but in this case, in time. This operation is characterized as increasing the amount and duration of fixations on parts of the object.

5. *Alternating Fixation* – This emphasizes comparative fixations of both objects. It is characterized as repeatedly fixating between the same parts of each object at least twice while the head mainly being stationary except for rotation.

6. *Alternating View* – This operation is similar to Alternating Fixation; however, it also includes a change of viewpoint (head location).

C. *Confirmation* – This stage acts as the control module to verify whether a potential strategy provides the correct answer.

1. *Strategy Repetition* – Interestingly, once a strategy has led to an answer, the subject sometimes repeats the strategy at least once. In other words, double or triple-checking the answer. From here it was also observed that the subject enters Strategy Formulation again to refine or dismiss the strategy.

D. There are a number of specific visual and spatial operations employed in the above strategies.

– Visual Operations describe the types of gaze movements and seem to be of two kinds.

1. *Tracing Connected Components* – This elemental operation is defined as a series of fixations spatially close together, for instance, following the object's silhouette.
2. *Comparing Arbitrary Components* – Instead of following a geometry with multiple fixations in close proximity, this elemental operation is characterized by more considerable distances between fixations, for example, observing faces of the object punctually.

– Spatial Operations describe the physical movement of the sensory apparatus from one spatial location to another and are also of 2 kinds.

1. *Point of View Change* – It is defined as moving the entire body to a different spatial location beyond the motion of the sensory apparatus (i.e., eyes).
2. *Viewing Angle Change* – Defined as moving the sensory apparatus only.

The italicized phrases just defined are the ones used in the directed graphs, shown in Figures 4.2 ***A***, 4.3 and 4.4, extracted for each trial and which summarize the full sequence of actions a subject exhibits. Previously, in Figure 2.2, a set of viewing directions (frusta) were shown without most of temporal links or annotations, nor functional interpretations. Figure 4.2 ***A*** is a "bird's-eye-view" figure (but can be downloaded and examined on a large screen as the caption details[1]) showing a single trial in order to give an impression of the scale of the solution space. This trial is what a subject did for the initial conditions of a pair of objects from the easy group of Figure 2.17, with an orientation difference of 90°, the objects, in fact, is the same, and the subject starting from the long position. The point of showing it in this form is to emphasize global characteristics. A portion (from t=3.40s and onward) appears in more readable form in Figure 4.3.

The first thing to point out is that subjects do not always take a direct path to a solution – paths are complex and made up of many strategies. One might wonder about these results since the target objects seem complex and perhaps not so realistic. However, in early pilot tests, the use of simpler structures, we observed that subjects quickly found a solution strategy that always led to the correct answer – count the number of blocks. To do so still required a number of fixation changes and a number of body and head movements, but the task was thus too easy and not instructive. We needed to push the limits of the task.

[1] `https://data.nvision.eecs.yorku.ca/active/action_seq.jpg`

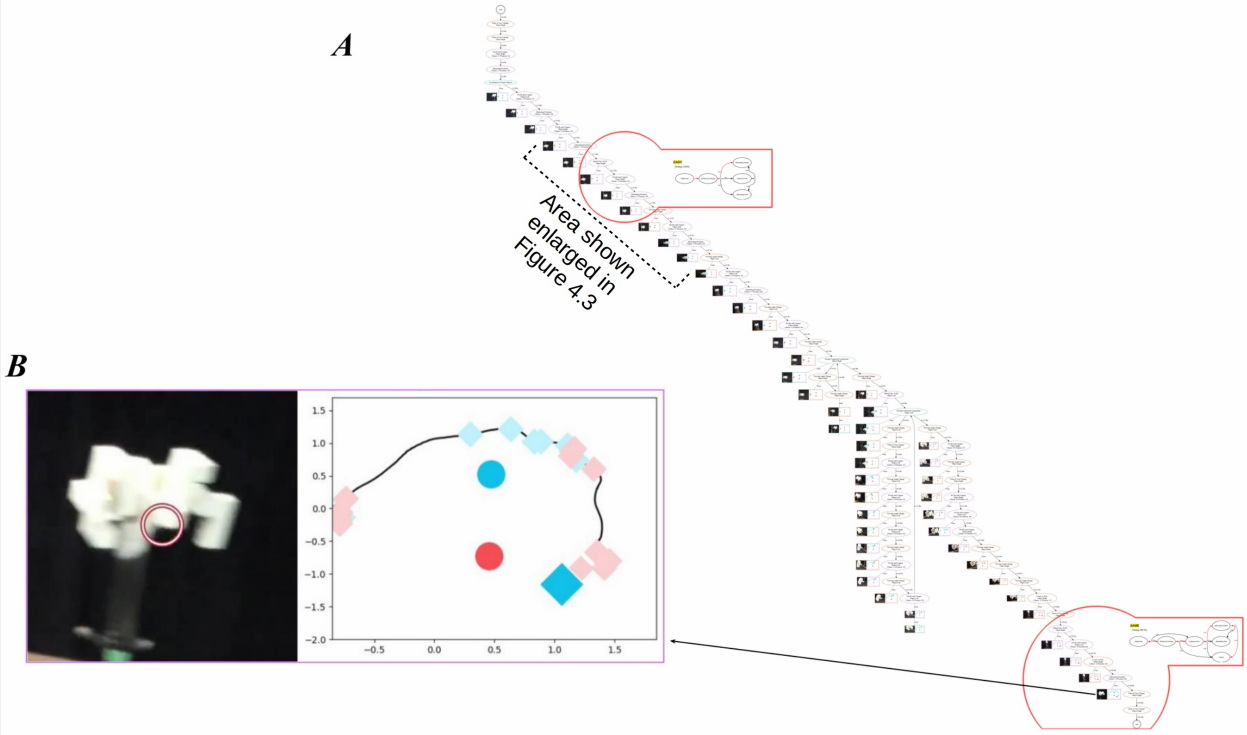


Figure 4.2: ***A***: This 'bird's-eye view' figure represents the sequence of actions taken by a subject for a pair of target objects from the easy group, with an orientation difference of 90°, the objects being the same, and the subject starting from the long position. This was the 9th trial this particular subject completed. The trial required 24.19s, and the subject performed 32 fixation changes with a total head movement of 10.05m. The final response was correct. The beginning of the trial is at the top. Two particular sub-graphs are highlighted with red circles (see Section 4.3.2 for more). The actual strategy identified is repeated on the right side of the red outline. Detail for the upper one can be found in Figure 4.4 (Easy). Note that illustrated branches are logical visualizations for strategies with defined start (branching) and end (returning to branch root) points, such as "Tracing Connected Components" and "Comparing Arbitrary Components." The area marked with a dashed line is shown enlarged in Figure 4.3. ***B***: Each of the dark blobs part A of this figure represents a particular gaze as shown here. On the left is the actual first-person camera-view with the red circle showing the point of gaze. The right portion shows the two target objects (red and cyan circles), the subject position (red diamond if viewing the red object, cyan otherwise) and the path traversed by the subject from the beginning to the current fixation. Smaller diamonds along the trajectory path indicate past fixations. The progression of this path is easily seen once the graph is magnified. The full resolution figure, best viewed with high magnification, is available at `https://data.nvision.eecs.yorku.ca/active/action_seq.jpg`.

A brief consideration of the scale of this problem in terms of the size of its potential solution space is instructive. As described, we observed an average 93 fixations during an average of 48s; this may encompass significant non-visual activity. With 300ms per fixation change, this leaves over 20 seconds for "acting" (moving the head or walking) and "thinking" (for instance, reasoning, planning,

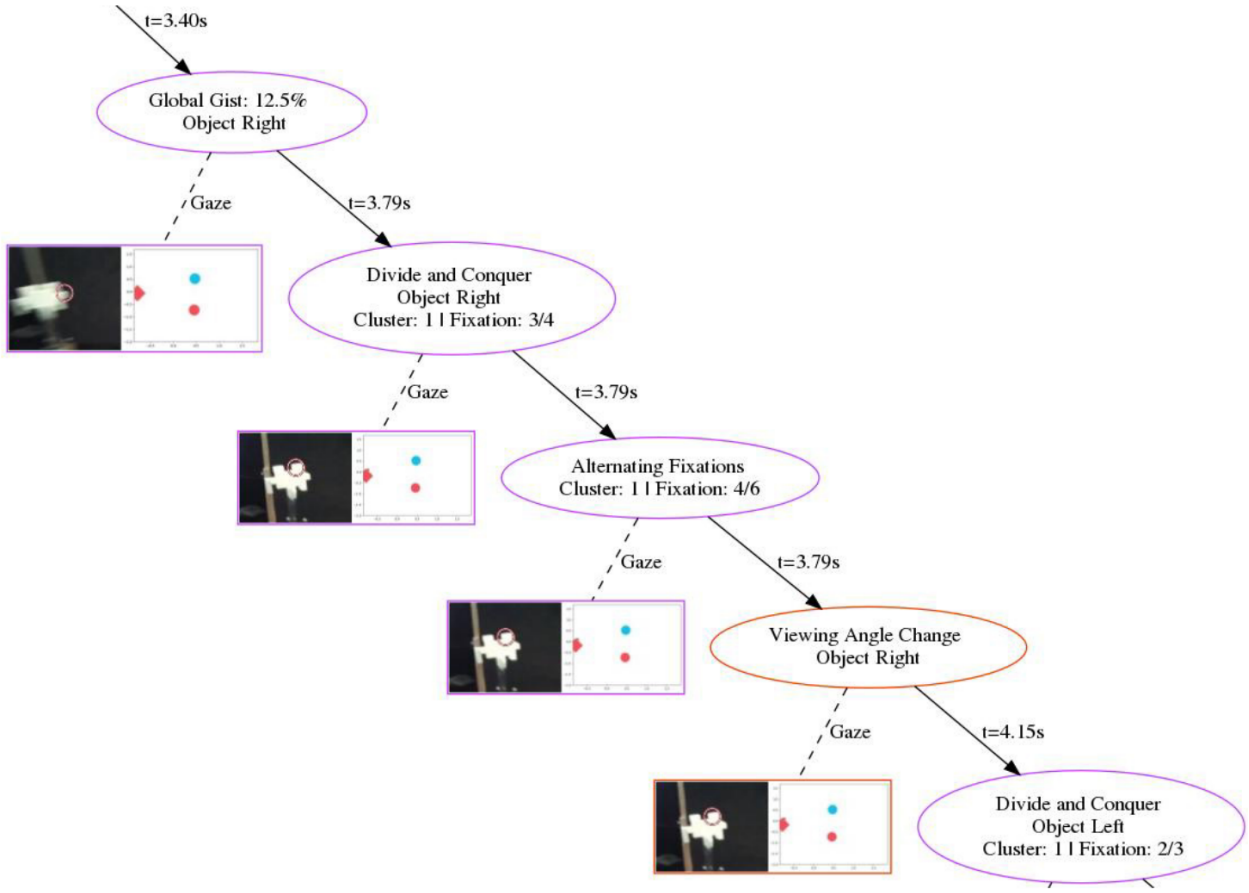


Figure 4.3: A zoomed portion of Figure 4.2 (marked with a dashed line) ***A*** beginning at the t=3.40s mark of the trial. As can be seen, most of the sub-elements include pictorial annotations of the actual action taken by the subject at that time as well as the actual point of gaze for that observation. The top three ovals comprise one path through the upper strategy outlined in Figure 4.2 ***A***, further described in, for example, Figure 4.4.

decision-making, working memory). The number of possible physical states of the experiment alone is enormous. Each fixation is defined by head pose (6DOF), fixation angle (the angle away from fully straight-ahead given head pose), and fixation depth (convergent binocular vision). The normal human visual field at the retina is $180° \times 100°$. At $2° \times 2°$ quanta (measurement error of $PESAO$), there are 4500 possible fixation angles. Similarly, the head may rotate left-right about a stationary body approximately another 180 degrees, and up-down another 180°. Say this is divided into 5° quanta, giving us 1296 possible head poses. Assume that the subject's body position may be located anywhere within the $4.3m \times 3.4m$ PESAO space; let's quantize this space into 0.4m x 0.4m units for convenience, so there are roughly 91 possible positions. A subject's body may be oriented in

any direction across 360° rotation, quantized in 5° units, this means roughly 72 possibilities. The physical states of the subject are the product of these numbers, over 3.8×10^{10}. With respect to the target space, the L_2 set has 12 objects, from which we choose 2 for each trial, and they may be the same. The choice of two is always from the same complexity subset, and there are three subsets. Subject starting position matters as it affects the spatial order of presentation. Add in 3 relative pose orientations, and we arrive at a total of 378 different targets starting configurations[2]. The space of sequences of subject and target configurations to cover the 6 to 800 fixations we have observed is more difficult to estimate but clearly is an unmanageably large number.

4.3 Cognitive Programs

As the experimental results presented in Section 3.4 and the examples in the above figures show, the sequence of observations humans use to solve the same-different task can be long and complex, perhaps unexpectedly so. That they seem dynamically composed, deployed and evaluated, dependent on the task type, complexity and initial conditions, with little learning effect, seems counterintuitive, especially when compared to modern computational attempts at active learning (however, see [Taylor et al., 2021] for review and a promising change). For the simple cases, our subjects showed simple, more routine methods, and for complex tasks, more involved strategies. Human intelligence is not designed for the simple case, but rather so that it can solve the most difficult as well. Simple solutions do not easily generalize, and our experimental design served us well in that it unveiled those situations.

Thus, a series of important questions arise: how do observers arrive at these strategies? Is the sequence determined all in advance, or is it constructed in parts, each depending on the result of the previous? How can these sequences be represented? Are portions of them learned in past experience and then used as needed? These human sequences are many seconds long, and thus the processing is clearly not all confined to the visual cortex, where one feedforward pass requires only about 150ms. What might these extra-visual actions be, how could they be inferred from the observed sequences, and how can they be expressed in algorithmic form? Steps towards answers

[2]For the *short* and *corner* starting positions, spatial positioning of targets matters, so there are 16 possible pairs of the 12 objects; for the long position because target objects are equidistant, there are only 10. The number of object configurations is computed for these two start positions separately and then summed.

follow.

4.3.1 Cognitive Program Concept

Cognitive Programs (CPs) are a modernized offspring of Ullman's seminal Visual Routines [Ullman, 1987] and provide an algorithmic structure to visual, cognitive, action and attentional behaviours. Ullman's proposal addressed how human vision extracts shape and spatial relations. He proposed that: visual routines (VRs) compute spatial properties and relations from base representations to produce incremental representations; VRs are assembled from elemental operations; new routines can be assembled to meet processing goals; universal routines operate in the absence of prior knowledge, whereas other routines operate with prior knowledge; mechanisms are required for sequencing elemental operations and selecting the locations at which VRs are applied; and, VR's can be applied to both base and incremental representations.

VRs have been studied and supported by both computer scientists and brain scientists and seem an enduring idea (see review in [Tsotsos and Kruijne, 2014]). However, the idea is dated in its detail and requires updating. Cognitive Programs [Tsotsos, 2010, Tsotsos and Kruijne, 2014] are an elaboration and modernization of the original idea and are based on a broader up-to-date view of visual attention and visual information processing. CPs are sequences of representational transformations, attentional tunings, decisions, actions, and communications required to take an input stimulus and transform it into a representation of objects, events or features that are in the right form to enable the solution of a behavioural task. Methods are like Ullman's universal routines, while when parameterized for the current situation, they are termed Scripts. In a real sense, CPs are algorithms for solving problems (see [Kotseruba and Tsotsos, 2017, Lázaro-Gredilla et al., 2019] for examples).

4.3.2 Mined Methods

On viewing the graph of Figure 4.2 $\boldsymbol{A}$, and remembering there are hundreds of these collected, one might wonder whether there are sub-graphs that are repeated and that perhaps they form standard chunks[3] of actions, in other words, the methods of our CP formulation. In fact, this appears to be

[3]The idea of chunking in cognitive science goes back to at least [Johnson, 1970]. The concept is used to group together elements of a behaviour sequence and link these to memory and their coding. Here we use the term method in keeping with the Cognitive Programs terminology introduced in [Tsotsos and Kruijne, 2014].

the case. We have used the sequential pattern mining approach *PrefixSpan* by [Pei et al., 2004] to detect such standard chunks. Other sequential pattern mining algorithms were tested as well, such as *SPAM* [Ayres et al., 2002], *SPADE* [Zaki, 2001], and *FAST* [Salvemini et al., 2011], using the data mining library by [Fournier-Viger et al., 2016]. However, the results of the different approaches were similar and varied mainly in the run time.

A total of 50 such methods represented by directed sub-graphs have been found, each with different usage frequency depending on target complexity, differences in target pose and subject starting position. All mined methods are shown in Figures 4.4 (object complexity), 4.5 (object orientation), 4.6 (starting point), and 4.7 (object sameness), all with respect to the measured independent variable (Section 2.4.9). Each of these combines 2 or more of the 50 methods into a kind of Bayes Net representation because they contain similar elements only with different frequency of observation. Decision points are shown as little circles in the graph with relative frequency usage labelled on the outgoing arcs. Instances of each are circled in Figure 4.2 ***A*** as examples (note that the top one is also part of the expanded view in Figure 4.3 so details can be seen).

Figure 4.4 shows the mined methods with respect to object complexity levels. For levels easy and medium, most frequently four methods (paths through graph) are deployed which have an occurrence of 100% and 99.2%, respectively. Most notably, both consist of identical strategies but with varying likelihoods. "Divide and Conquer" always follows "Global Gist" and the remaining three options are selected in a similar order. However, if "Coarse to Fine" is chosen – for both cases with a probability of 37% – what is chosen next varies. For easy cases, it is more likely to use "Alternating Fixation" over "Alternating View". For medium cases, this changes; "Alternating View" is more likely than "Alternating Fixation". While all probabilities are close – around the 50% mark – a possible explanation to this is that with increasing complexity levels, subjects need to move more to observe all necessary details of the objects, hence the change to "Alternating View". Remember, "Alternating View" includes head movements of at least 60 cm, which allow changing what is being observed quite a bit.

For the hard case, again, four methods are most dominant and can be observed in 84.3% of all trials with this set up. However, in contrast to easy and medium, "Coarse to Fine" is always included. This means that subjects deal with the hard complexity level by revisiting the same part of the object but with an extended time of fixation.

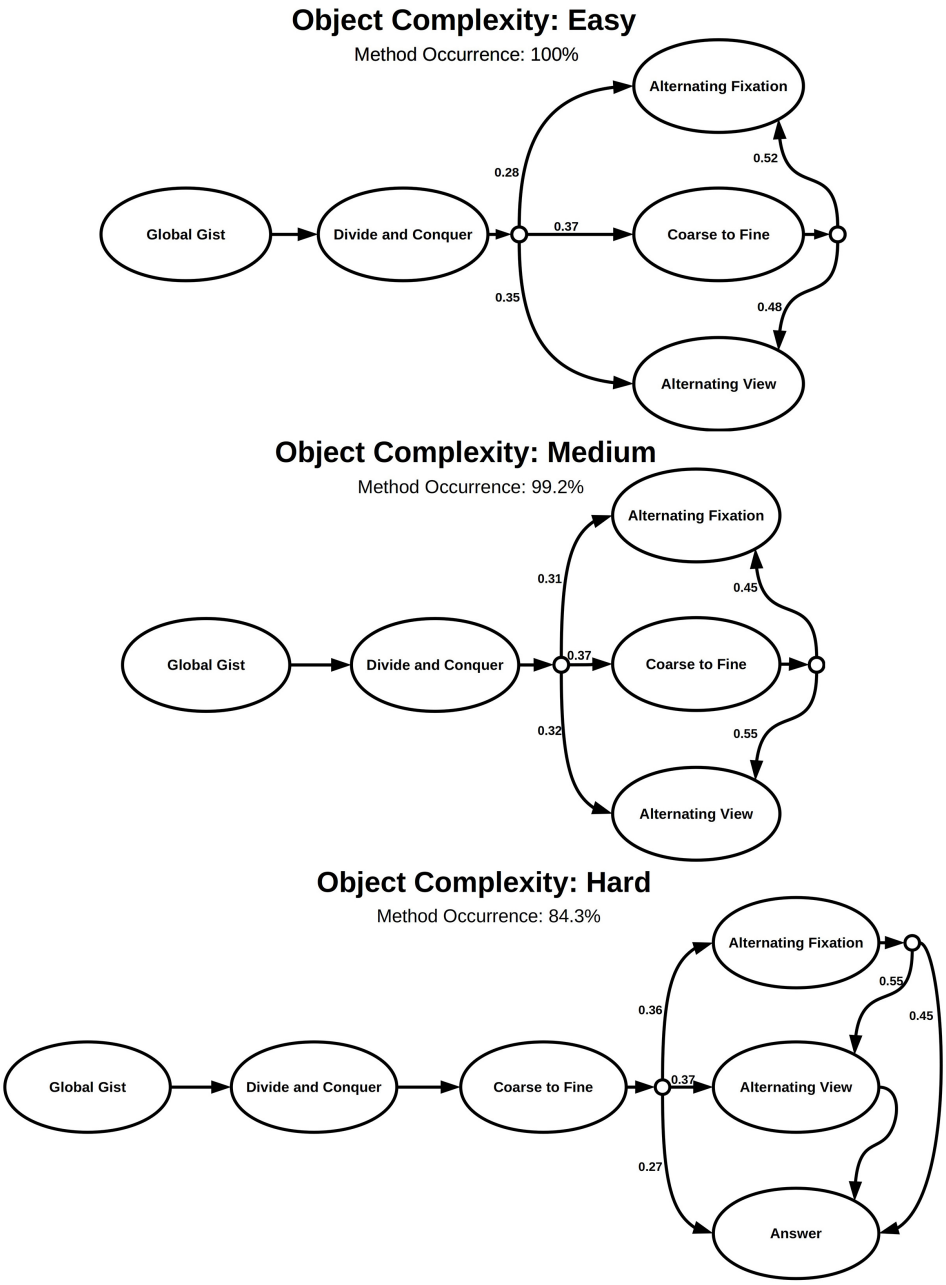


Figure 4.4: Mined methods with respect to *object complexity*. The labels on the arcs between nodes give the frequency of occurrence for that arc. The small circles are choice points. Top: This sub-graph of actions appears in all trials that involved easy objects (occurrence: 100%). Middle: This sub-graph of actions was found in 99.2% of all trials that involved objects of medium complexity. It covers 4 of the 50 methods found (4 different pathways through that graph). Bottom: This sub-graph was found for objects with hard complexity. The bottom right node is labelled "Answer", and this means that this sub-graph was found as the last sub-graph in the sequence for these cases.

In comparison to the methods found for object complexity, methods for orientation difference are more diverse; method occurrences do not range from 84.3% to 100%, they range from 71.6% to 78.5%. Figure 4.5 illustrates in total 10 methods (2 for 0°, 3 for 90° and 5 for 180°). While for 0° and 90° the first three strategies are the same ("Global Gist", "Divide and Conquer", and "Coarse to Fine"), for 180° no such guaranteed sequence exists. It is a 50:50 change whether "Divide and Conquer" follows "Coarse to Fine" and similarly whether "Coarse to Fine" or "Alternating View" follows "Divide and Conquer". In other words, subjects approached this setting with different sequences of strategies. However, all of these end in the terminal state "Answer". While for 0° and 90° most sequences, except for the one from 90°, occur throughout a trial.

Mined methods for different starting positions show an increase of variability in strategies for two of the three settings ("Corner" and "Long"), unlike the variables we have discussed so far. Figure 4.6 shows the methods. The same strategies are part of these methods as before. In fact, all three settings consist of the same strategies. The method occurrences range from 72.2% to 76.8%. However, for the first time, the methods for varying starting positions include the terminal state "Answer" for all graphs. This means that there is a larger agreement on how to approach the "Answer" state in comparison to the other tested variables, including the "sameness" variable, as we will see next.

The effect of "sameness" on the mined methods is illustrated in Figure 4.7. Similar to other variables, these methods consist of the same strategies. However, only the setting for the same objects, consists of four methods that end in the terminal state "Answer", whereas the setting for different objects does not have any commonly used sequence of strategies (method) that ends in the terminal state. This shows that the final sequence for the case that the objects are different vary among the subjects so much that no common pattern can be mined.

Mined methods, although different in content, are like the CPs described in [Tsotsos and Kruijne, 2014] or in [Kotseruba and Tsotsos, 2017]. For example, *Global Gist* is an abstract concept that, as described above, tracks and controls how much of the virtual viewing sphere around each object has been explored. *Coarse-to-Fine* is similarly abstract and is characterized as increasing the amount and duration of fixations of parts of the object. Each involves complex internal actions on its own. In other words, it is expected that these methods will have an internal structure, which itself may be a composition of other methods. At their most primitive, some methods may be

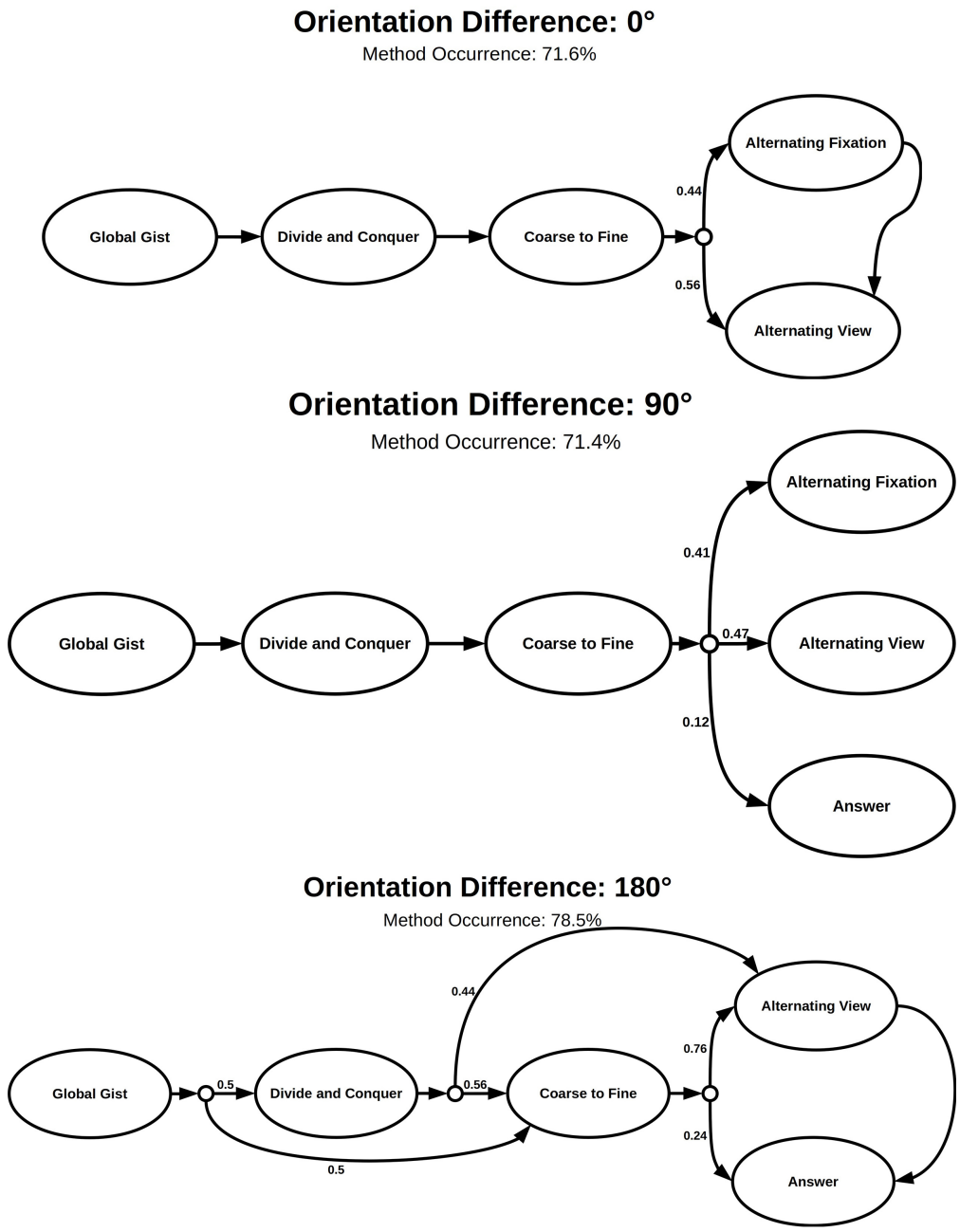


Figure 4.5: Mined methods with respect to *orientational difference*. The labels on the arcs between nodes give the frequency of occurrence for that arc. The small circles are choice points. Top: This sub-graph of actions appears in all trials that involved an orientation difference of 0° (occurance: 71.6%). Middle: This sub-graph of actions was found in 71.4% of all trials that involved an orientation difference of 90°. Bottom: This sub-graph was found for objects with an orientation difference of 180° (occurance: 78.5%).

innate, but others would be learned.

Figure 4.8 illustrates what might be considered such an internal structure for "Alternating

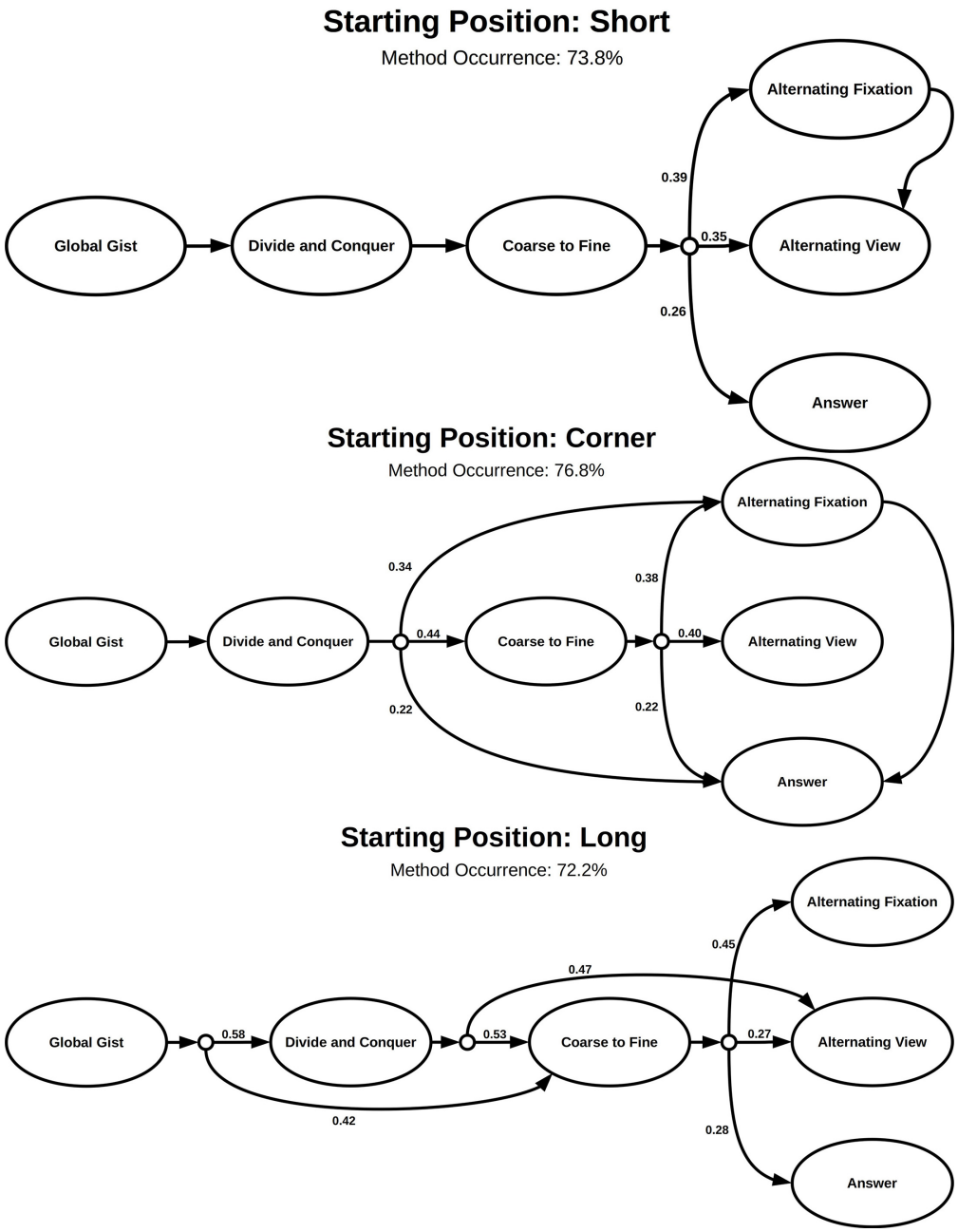


Figure 4.6: Mined methods with respect to *starting position*. The labels on the arcs between nodes give the frequency of occurrence for that arc. The small circles are choice points. Top: This sub-graph of actions appears in all trials that involved the starting position on the short side (occurrence: 73.8%). Middle: This sub-graph of actions was found in 76.8% of all trials that involved the starting position in the corner. Bottom: This sub-graph was found for the starting position on the long side (occurrence: 72.2%).

Fixation", defined in Section 4.2, and a component of both CP's in Figure 4.4. This is extracted from an actual trial and shows gaze moving from one object to the other, seemingly examining a

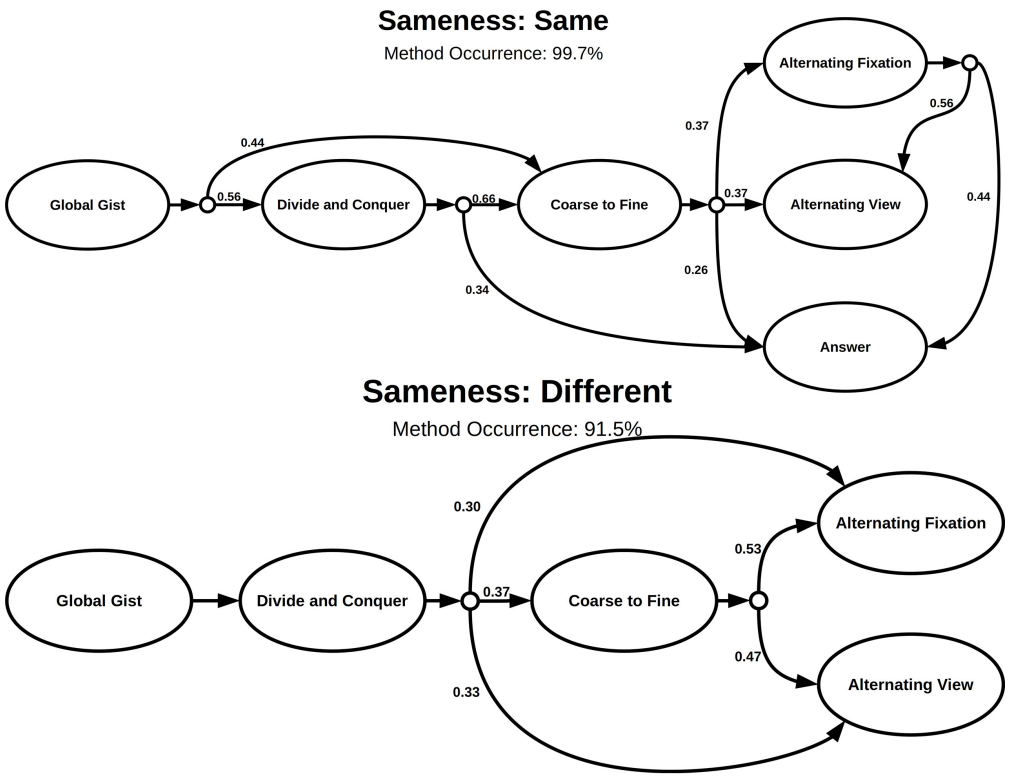


Figure 4.7: Mined methods with respect to *sameness*. The labels on the arcs between nodes give the frequency of occurrence for that arc. The small circles are choice points. Top: This sub-graph of actions appears in all trials that present the same objects (occurance: 99.7%). Bottom: This sub-graph of actions was found in 91.5% of all trials that involved different objects.

small portion of the physical structure of each object, presumably to determine if that particular section was the same or different.

Figure 4.9 shows an illustration of "Divide and Conquer" generated from our data. The subject revisits the same area of an object multiple times. However, this strategy divides the object into smaller parts. Here, the subject observed a large area of the object first with multiple fixations and narrowed down in two additional observations to a single connecting point of two elements of the object (Figure 4.9 left to right).

Lastly, an illustration for "Global Gist" is shown in Figure 4.10. Here a bird's-eye-view of the sectors surrounding one of the objects is shown. This strategy divides the virtual viewing sphere into eight sectors, as defined in Section 4.2. We show one real instance in which the subject started its first fixation in Sector VII moved to Sector I to observe another aspect and ended the trial in Sector II. Shown here are only the first fixations of each sector. In between, more fixations were

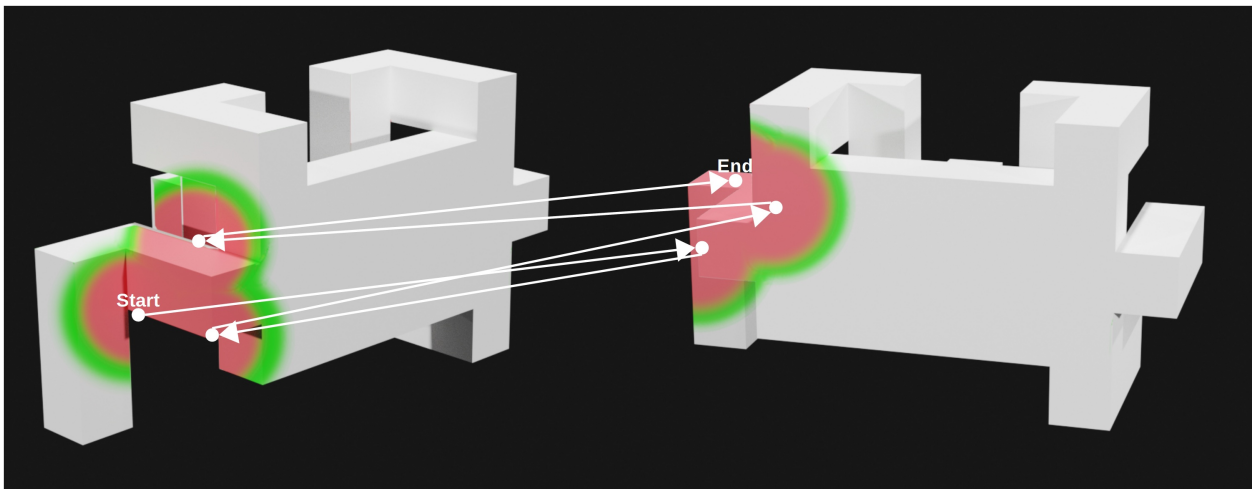


Figure 4.8: Here, we show a visualization of a fixation pattern that we identify as *Alternating Fixation* generated from our data. Both objects are displayed in the orientation of their observation. The corresponding fixations are highlighted with red circles and a green border. Arrows point to and originate at the center of gaze. Further, the starting fixation is provided (annotated with "Start"), and the subsequent fixation is connected with an arrow. The alternating fixation ends at the fixation marked with "End." The two objects are of complexity medium; they are the same object, presented at 90° orientational difference. The mean accuracy for gaze fixations is 1.42°. Color encoded with uncertainty boundary in green.

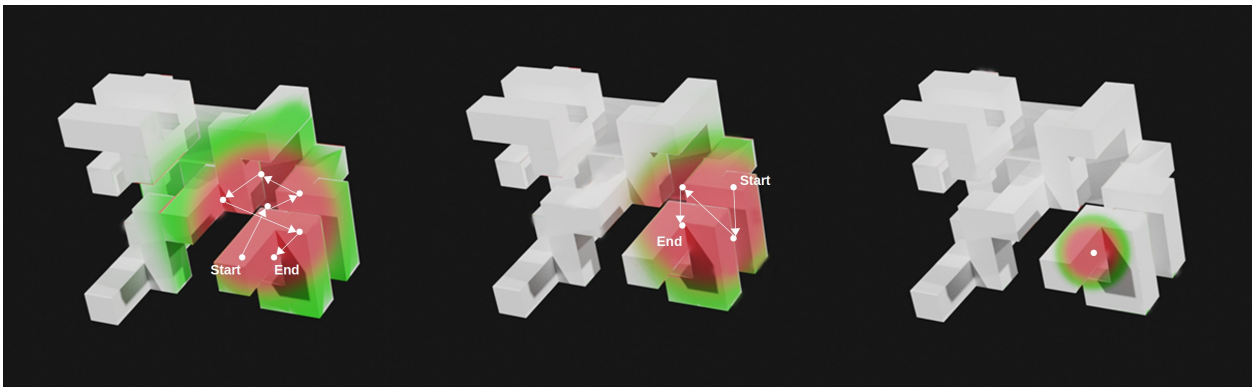


Figure 4.9: Here, we show a visualization of a fixation pattern that we identify as *Divide and Conquer* generated from our data. The corresponding fixations are highlighted with red circles and a green border. From left to right, three observations of the same area of the object are shown. The first observation consists of seven fixations (centers are marked with a white circle), covering a larger area (fixations are connected with arrows) which is then more and more (four fixations) refined until only a single connecting point of two elements of the object is observed (one fixation). The object is of complexity hard, the second object was the same, presented at 90° orientational difference. The mean accuracy for gaze fixations is 1.42°. Color encoded with uncertainty boundary in green.

executed; in sector VII a total of 11 fixations were recorded, in sector I four fixations, and in sector II six fixations. This trial was of object complexity medium, the different objects were shown, and

they were presented at 180° orientational difference.

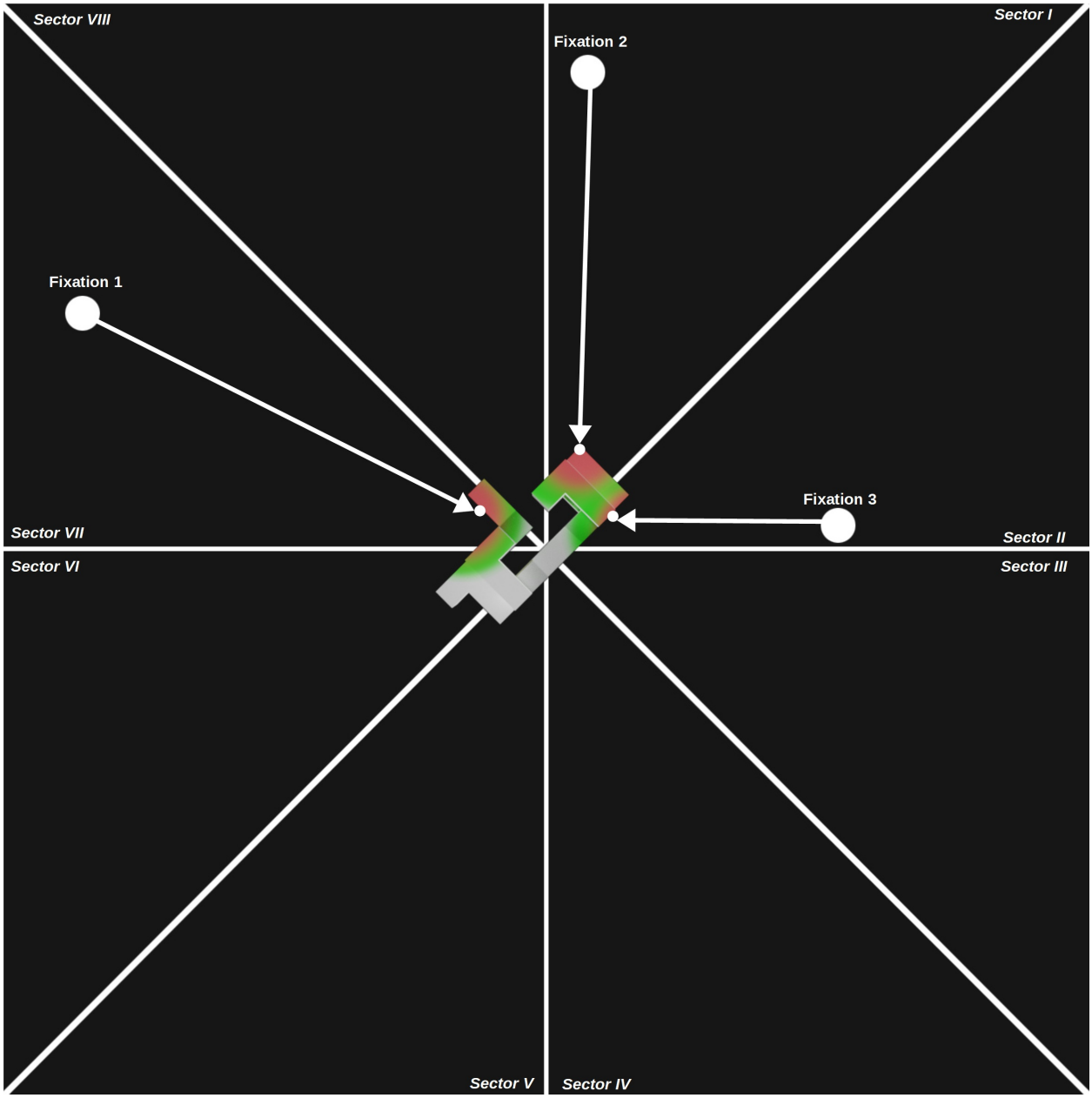


Figure 4.10: Here, we show a visualization of a fixation pattern that we identify as *Global Gist* generated from our data. A bird's-eye-view of the sectors surrounding one of the objects is shown. The corresponding fixations are highlighted with red circles and a green border. Arrows point to and originate at the center of gaze. Only the initial fixation for each section is illustrated for simplicity. However, a total of 26 fixations were recorded for this trial. This trial was of object complexity medium, the different objects were shown, and they were presented at 180° orientational difference. The mean accuracy for gaze fixations is 1.42°. Color encoded with uncertainty boundary in green.

As described in the Introduction, we consider the visual system as a general-purpose processor,

tuned to the task and input of the moment. It might be that you have learned thousands of such CPs, and you have them stored in memory, quickly deploying the right one(s) at the right time, dynamically parameterized for the task. This might suggest the need for a controller of some sort, and although there is controversy about such a need as [Tsotsos et al., 2021] describes, both theoretical and experimental arguments in that chapter point to the clear importance of a controller. Some computational structures must be present to accept task instructions and translate them into executable CPs for solving the task. The timings of these actions need synchronization and coordination. How exactly this might occur is too much for the scope of this chapter. However, the feasibility of the idea has been documented not only in the earlier VR work but also in our more current CP work [Abid, 2018, Kunic, 2017, Kotseruba and Tsotsos, 2017, Tsotsos and Kruijne, 2014]. [Abid, 2018] used CPs to represent and test human behaviour for several attentional psychophysical experiments. [Kunic, 2017] developed a compiler that would accept Imperative English and provide task specification for a CP while in [Kotseruba and Tsotsos, 2017], two video games were encoded and tested to human top rank performance. The current results lend further support to CPs' potential for complex real-world tasks.

4.4 Summary

In this chapter, we have explored and identified the characteristics of human behaviour in active tasks that could inform the development of cognitive systems that perform similar tasks. Obvious problem instances were simply solved, but as we increased the difficulty, the apparent simplicity of the task masked the surprising complexity of human strategies deployed.

Here, we present our results, while data collection and analysis continue. Among the most striking of the results are the following:

- Fixation sequences seemed purposeful both as reported by subjects and by examining the sequences. We were able to identify 50 patterns of actions - *Cognitive Program methods* - that were repeated in various combinations in all trials.
- The CP methods as described must be considered as simplified. That is, each of the nodes in the sequence represents yet another sequence of actions, including less abstract computations

such as image analysis, attention, use of working memory, decision-making, planning and re-planning, etc.

This particular task seems an excellent testbed for testing systems that purport intelligent behaviour. The Turing Test, just as an example, does not test active observation in the ways our task requires. There is no claim that this should replace it, of course; nevertheless, our data does point to a dimension of intelligence - the ability to decide how, why, when, what and where to sense the environment to best complete a task – that has not been well studied.

We have proposed that Cognitive Programs provide a flexible, dynamic composition of potential solutions. We intend to also experiment with three-dimensional "spatial relations", three-dimensional "visual search", and to add shadowing to all the tasks within the *PESAO* facility. The goal is not to solve each separately but rather to discover the common elements of a generic visual problem-solving strategy. The reality of active human behaviour will likely reveal many surprises to come, and with that, many challenges for how artificial agents may be developed with the same abilities.

We discovered that humans exhibit a variety of problem-solving strategies whose breadth and complexity are surprising and not easily handled by current methodologies. The importance of active observation is striking. These results highlight the new dimensions of visuospatial problem-solving that active observers employ.

Exactly how the problem-solving strategies are combined and selected is beyond the scope of this work but opens an exciting avenue of research. In fact, an avenue which will be investigated in follow-up work.

In the next chapter, we present our approach to learning the *Three-Dimensional Same-Different Task for Active Observers* with a modern machine learning method.

Chapter 5

Learning Active Visual Behaviours

5.1 Introduction

The findings presented in Chapter 3 and 4 raise the question of how they might inform strategies used by modern machine learning methods.

In this chapter, we present our approach to learning the same-different task with a modern machine learning method. We provide an overview of the original goal of Artificial Intelligence (AI) (Section 5.1.1), what the current goal of AI is (Section 5.1.2), and why the original goal should not be completely neglected (Section 5.1.3). We move on explaining why reinforcement learning is best suited for the same-different task and continue with a brief introduction to reinforcement learning (Section 5.2), present our efforts to learn this task (Section 5.3), provide an interpretation of results (Section 5.4), and lastly in Section 5.5 we conclude this chapter.

5.1.1 The Original Goal Of Artificial Intelligence

Dating the beginning of AI is difficult. The idea of inanimate objects coming to life as intelligent beings goes back thousands of years. The ancient Greeks had myths about robots (See, for example, the story about Talos [Woodcroft and Others, 1851] in Section 1.1), and ancient Chinese [Loewe and Shaughnessy, 1999] and Egyptian [Maspero, 1895] engineers built automatons. It is also said that Leonardo Da Vinci presented a "robotic knight" at the court of Milan in 1495 [Rosheim, 2006]. Figure 5.1 shows a recreation of said robot. Theoretically, the beginnings of modern AI can be traced back to the attempt of classical philosophers to describe human thinking as a logical system [Ackrill, 1975]. Formally, the field of AI was not founded until the Dartmouth Summer Research Project of 1956 that initiated this research discipline [Moor, 2006]. John McCarthy is credited for giving the field its name.

Initially, McCarthy was disappointed that the papers in automata studies did not cover more about the possibility of computers having intelligence. On August 31st, 1955, together with Marvin L. Minsky, Nathaniel Rochester and Claude E. Shannon, McCarthy presented a proposal for the Dartmouth summer research project on artificial intelligence [McCarthy et al., 2006]. They proposed a "2 month, 10 man study of artificial intelligence during the summer of 1956". Specifically:

> "The study is to proceed on the basis of the conjecture that every aspect of learning and any other feature of intelligence can in principle be so precisely described that a

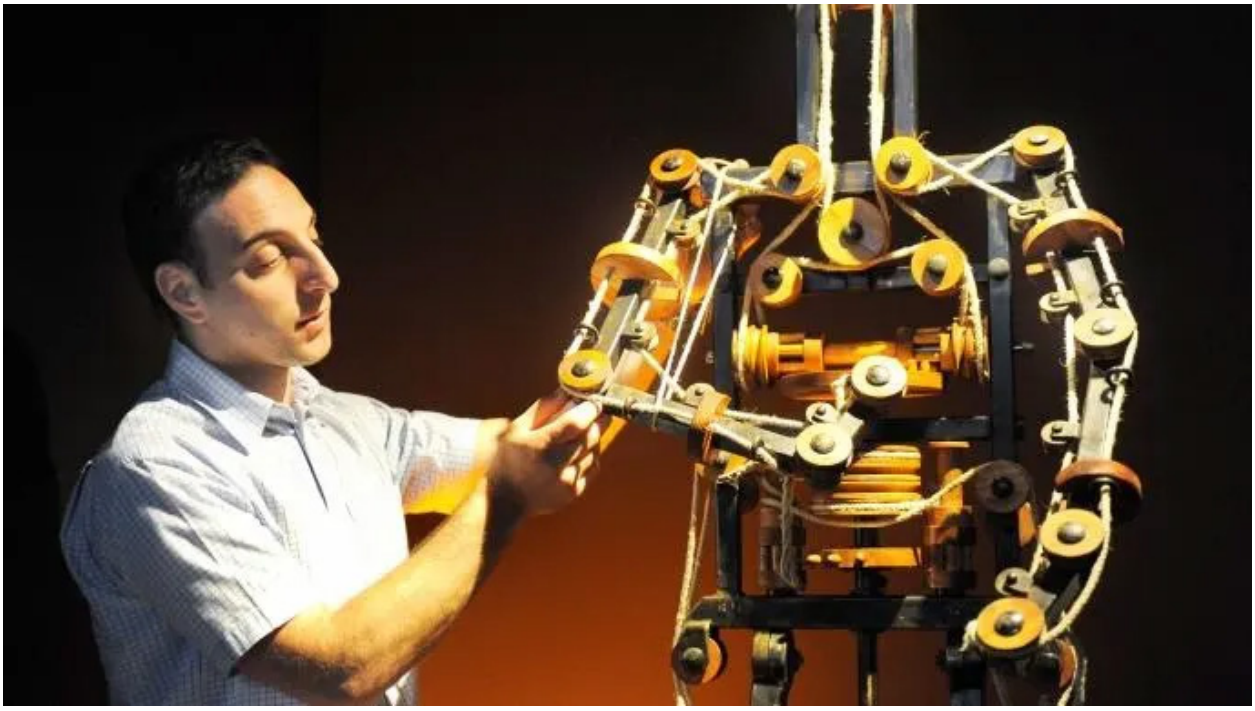


Figure 5.1: Life-sized recreation of Da Vinci's "robotic knight". It is said that the robot was first displayed by Da Vinci at the court of Milan in 1495. Credit: William West/AFP/Getty Images.

> machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of *problems now reserved for humans*, and improve themselves."

Most of modern AI research, however, goes back to the Turing Test [Turing, 1950b]. The test, originally introduced as the *imitation game* by Alan Turing in 1950, is a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human. Figure 5.2 illustrates the standard interpretation of this test. It was proposed that a human evaluator would judge natural language conversations of a human and a machine. The conversation is limited to a text-only channel, so the evaluator does not know who the human and machine are. If the evaluator is unable to tell the machine from the human during a 5 minutes interaction, the machine has passed the test. The test does not measure if the machine provides correct answers, more so if the answers resemble those a human would give. The emphasis of this test, contrary to Minksy's and other's AI goal, is on input-output-behaviour, also known as external behaviour, with no concern for what happens internally, for instance, how the answer was generated, what the sequence of steps is to produce the answer, and so on. In contrast to the original goal, the Turing Test is limited to correct

input-output behaviour – it tests how natural a "sequence" of sentences might be.

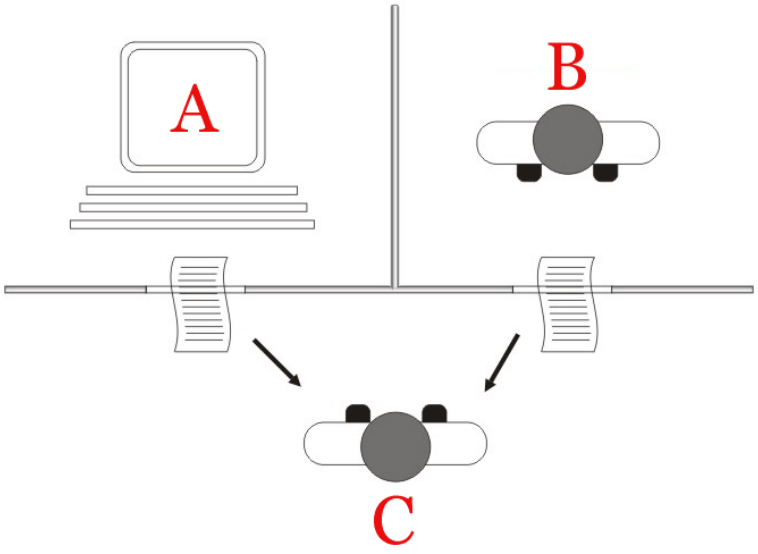


Figure 5.2: The standard interpretation of the Turing Test. C, the human evaluator, is asked to decide whether A or B is a human. The C is only allowed to use responses to written questions. Making this mainly a test about input-output behaviour. Source: `https://commons.wikimedia.org/wiki/File:Test_de_Turing.jpg`.

5.1.2 Today's Artificial Intelligence

The majority of the research community that focuses on AI follows the inspiration of the Turing Test, therefore the development of intelligent input-output behaviours. The currently dominant AI research method with respect to this thesis is machine learning. It addresses the question of how to build computers that are able to improve through experience automatically [Jordan and Mitchell, 2015]. Recent progress is driven by advances in the development of learning algorithms, the theory behind them, the ongoing explosion of available data, and faster computation. Machine Learning methods have been adopted in a wide range of applications.

Essentially, the methods can be divided into supervised, unsupervised and reinforcement learning classes [Mitchell, 1997, Jordan and Mitchell, 2015]. Figure 5.3 provides an illustration of different Machine Learning methods and their specializations.

Supervised learning methods [Hastie et al., 2005] are the most widely used. Generally, these methods exemplify the function approximation problem in which the training data has the form of a collection of (x, y) pairs, where x is the data (for example, the image of a dog) and y is

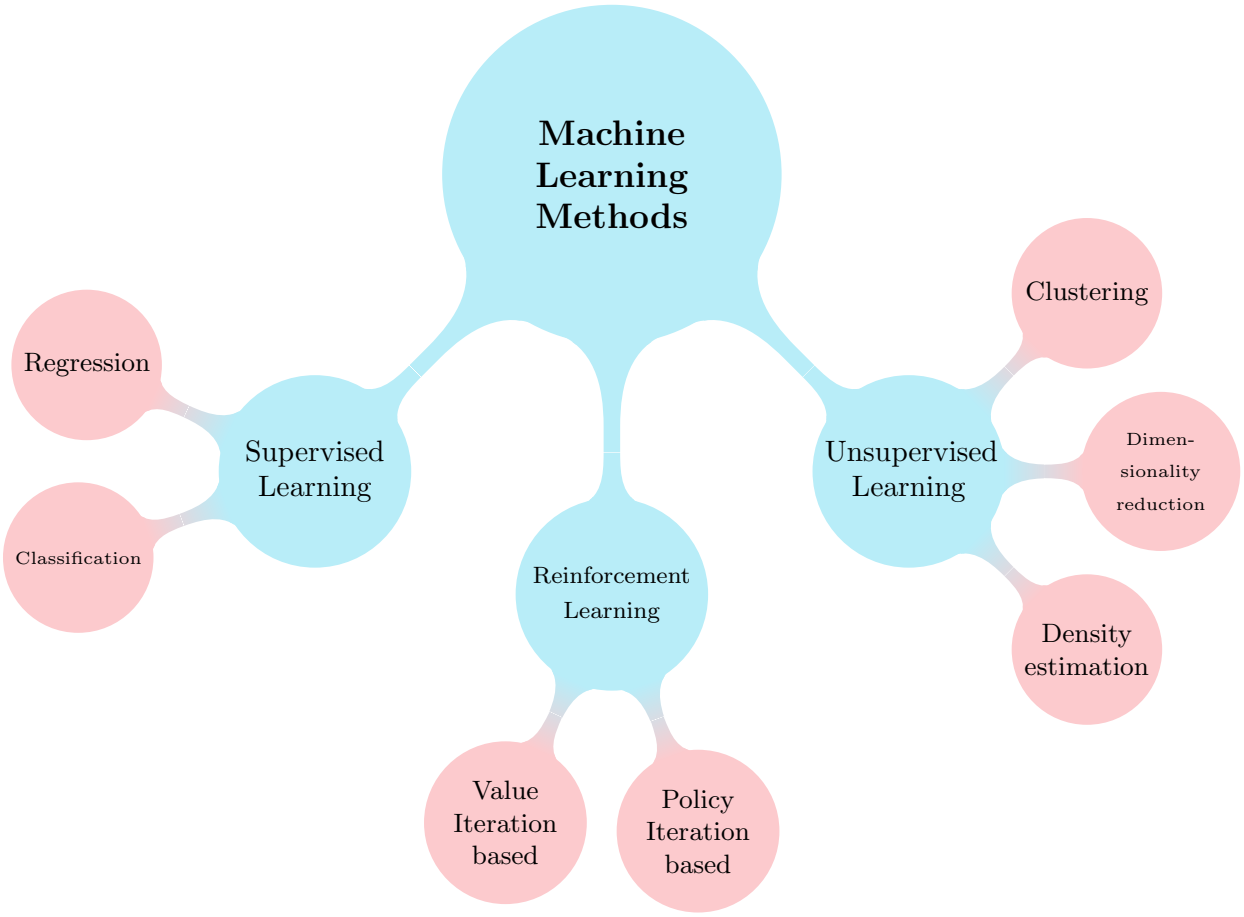


Figure 5.3: Diagram of Machine Learning Methods

the label (for example, the breed of a dog, such as *Portuguese Water Dog*. Figure 5.4 shows an illustration.). The task is to produce a prediction y^* in response to a query x^*. These methods form their predictions via a learned mapping $f(x)$, which produces the output y (or a probability distribution over a selection or all possible y) for the input x.

Common applications are regression and classification. In short, the difference between regression and classification algorithms is that the former predicts continuous values such as prices, coordinates, etc. and the latter is used to predict discrete values such as class labels.

Figure 5.4: Supervised learning method recognizing a portuguese water dog in an image.

Unsupervised learning methods [Murphy, 2012] describe algorithms that discover useful representations of the input without the need for labelled training data [Hinton and Salakhutdinov, 2006], unlike supervised learning methods. These methods involve the analysis of unlabeled data under different assumptions about the structural properties of the data, for instance, algebraic, combinatorial, or probabilistic [Jordan and Mitchell, 2015]. Applications for unsupervised learning methods are clustering, dimensionality reduction, and density estimation.

Lastly, the third primary machine learning paradigm is reinforcement learning [Sutton et al., 1998, Kaelbling, Leslie Pack and Littman, Michael L and Moore, 1996, Kober et al., 2014, Arulkumaran et al., 2017]. As Figure 5.3 suggests as well, the available information is intermediate between supervised and unsupervised learning. Here, the training data does not provide an indication of whether the output, given an input, was correct or incorrect. The training data in reinforcement learning only provides an indication (reward) as to whether the chosen action of the agent is correct (positive reward) or not (punishment/negative reward). In comparison to the other two machine learning paradigms, if the action is incorrect, there remains the problem of finding the correct action. Furthermore, reinforcement learning includes an active component that lets the agent interact with the environment (more in Section 5.2); hence in some way, it has to make decisions. Lastly, it provides a method to program agents by reward-punishment without needing to specify *how* the task is to be achieved [Kaelbling, Leslie Pack and Littman, Michael L and Moore, 1996].

Roughly speaking, reinforcement learning algorithms can be divided into value-iteration-based and policy-iteration-based methods. The difference is that the former starts with a random value function, whereas the latter starts with a random policy (control strategy).

Although it is helpful to use these three paradigms to organize Machine Learning methods, blends across these categories exist. For instance, semi-supervised learning uses unlabeled data to augment labelled data in a supervised learning context. In their standard interpretation, all three methods have a common emphasis on learning input-output-behaviour with large amounts of data.

5.1.3 The Path Between Input and Output Matters

In contrast to the modern machine learning approaches described before, sub-areas of the AI community working on cognitive architectures or computational neuroscience do put an emphasis on modelling priors. Related to this, in the field of cognitive architectures, the term used is *internal factors* (which includes motivations, affective states, emotions, moods, drives, and others) and describes the selective bias to determine the next step [Kotseruba and Tsotsos, 2020].

It is fair to say that in some cases, priors are not crucial to the problem at hand. However, for instance, if we want a child to learn how to build a configuration of blocks and we close our eyes between start and finish, we cannot point out which intermediate step led to an incorrect output. Thus, learning becomes trial and error only, not purposeful – certainly not how a human teacher

would function. Knowing the path between input and output is important in scenarios in which the system's behaviour needs to be explainable, anticipatory or generally speaking, understandable to humans.

One such field is Human-Computer Interaction (HCI). It describes the multidisciplinary field of study focusing on the design of the interaction between humans and computers. It is especially interesting to build systems that are intuitive to use for humans. This goes beyond explainability, of course, as it also covers human psychology, emotional design and more. However, the system needs to be human-like in order to make the experience as personal as possible.

A special form of HCI that marries it with robotics is Human-Robot Interaction (HRI). Here, the computer is embodied and can take the form of stationary robots, like industrial collaborative robots and mobile robots, or like autonomous mobile robots for material handling. Application areas go beyond industrial robots and cover medical robots, social robots, automatic driving, search and rescue, and space exploration. A focus of this field is how humans and robots may better collaborate [Huang and Mutlu, 2016]. The dominant social cue for humans while collaborating is the shared perception of an activity. Anticipatory control allows the robot to proactively perform task action based on anticipated actions of their human partners [Robla-Gomez et al., 2017]. This is a vital ability for any robotic system whose role it is to be a real assistant at home, manufacturing, service or medical setting. [Huang and Mutlu, 2016] proposed a system of "anticipatory control" which enables robots to proactively plan and execute actions based on the anticipated human partner's task intent. The task intent is inferred from the gaze. Figure 5.5 shows the experimental setup, as well as involved methods. [Sheridan, 2016] points out that "all robots for the foreseeable future will be controlled by humans, either as teleoperators steered by continuous manual movement or as telerobots intermittently monitored and reprogrammed by human supervisors." He goes to say that it is a major human factors challenge to address how humans and robots need to have mutual models of each other. Humans that supervise robots need to have the predictive ability to intervene when needed. Humans working with robots need the predictive ability to know what next action should be next. For robots assisting humans, the human behavior they observe must be understandable and expected according to their internal models and plans, especially for populations that may have challenges or in difficult scenarios and environments.

To achieve this, it is crucial to building systems whose behaviour is *human-like* – in other

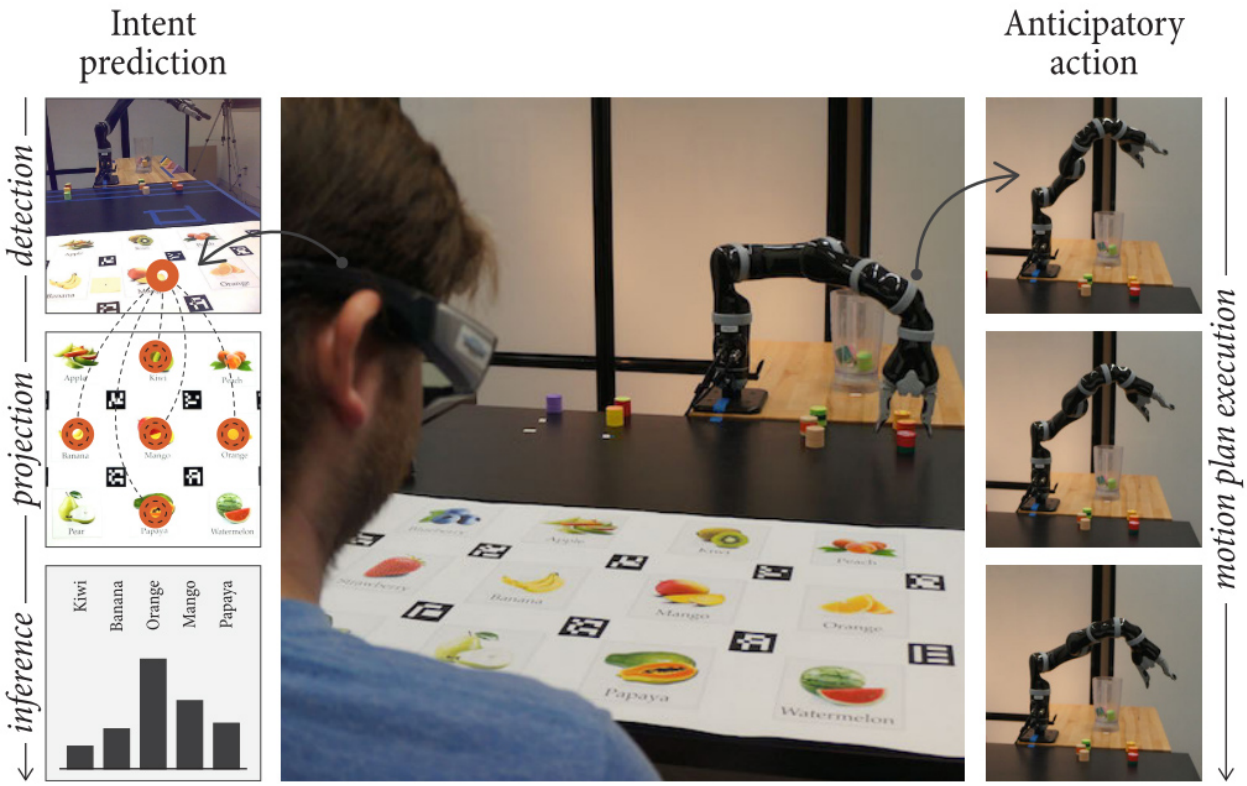


Figure 5.5: The experimental set up of the "anticipatory control" method which enables robots to proactively plan and execute actions based on the anticipated human partner's task intent. The task intent is inferred from the gaze. Source: [Huang and Mutlu, 2016].

words, so a human would recognize the behaviour as made by a human. This is similar to the aforementioned Turing Test; however, it does not only require the end-result to be human-like but also the steps in between. Let us consider an assistive home robot whose role it is to help around the kitchen, for instance, loading the dishwasher. While it is important that the dishwasher is loaded correctly – to the user's expectation – there are many steps involved in between, such as approaching dishes, cutlery and utensils, grasping them, moving them over into the dishwasher, and so on. Such a decomposition can be thought of as a hierarchy of functions (similar to what our subject did when they approach the same-different task), for instance, as seen with problem solvers like *STRIPS* [Nilsson and Fikes, 1971]. Especially when it comes to sharp knives, one does not want that these are handled unpredictably. One of our goals is to be closer to the original goal of AI and build systems whose steps between input and output are human-like. Thus, one could predict what happens next, intervene if needed, understand why any errors are made, and so on.

Hence, our definition of *human-like* goes beyond correct input-output behaviour as proposed

with the Turing Test. It also includes that similar, human-like,

1. steps are taken to achieve the goal
2. amount of data is that needed
3. time to learn is needed
4. error rates are exhibited
5. kind of errors not only at inference but also during training are seen

To the best of our knowledge, no approach in the field of computer vision exists that aims at solving all of these characteristics in their entirety. However, approaches exist that address one or some of them. For instance, the work of [Ognibene and Demiris, 2013, Huang and Mutlu, 2016] deals with anticipatory robot control and focuses on making the execution understandable and predictable for humans using visual perception to predict the next action of the human, hence presenting an example for item 2 of the list. Other examples can be found in the field of humanoid robots in which the goal is to build robotic systems that emulate human locomotion [Andreopoulos et al., 2011].

The computer vision community considers zero-shot [Larochelle et al., 2008, Lampert et al., 2009], one-shot [Lake et al., 2011] and few-shot [Fei-Fei et al., 2006, Fink, 2004] learning approaches to reduce the problem of immense amounts of data to learn a specific task. An example for a one-shot learning task is shown in Figure 5.6. These approaches fall under the realm of *transfer learning* and therefore still require in total a vast amount of data. Methods are trained first on a large dataset and then transferred to a different one. Usually, the different dataset is from a somewhat related task, but using less data. Specifically, zero-shot approaches does not use any additional data, one-shot with one example of the new task, and few-shot with a small amount of data, just as the names would suggest. These approaches show a step towards our third characteristic but do not fulfill it as they still require a large amount of data to initialize the method. However, it is an exciting direction of research.

For the fourth characteristic, most attempts to limit the time for learning are usually only indirectly addressed with faster hardware when it comes to modern machine learning approaches

Figure 5.6: One-Shot learning example to test yourself. An example is given in the red box. Can you find the others in the array? Source: [Lake et al., 2011]

[Krizhevsky et al., 2012]. However, exceptions exist besides architectural considerations, such as reducing the input resolution, removing input channels, using fewer layers, etc. Some approaches point the network towards the region of interest by pre-processing the input. For instance, SIFT features [Lowe, 2004] in combination with the optical flow is used for real-time human detection for aerial applications [Aldahoul et al., 2018]. In another approach, the discrete Fourier transformation is used to speed up learning [Highlander and Rodriguez, 2016]. Empirical results show that their method is able to reduce computational time by a factor of up to 16.3 times compared to traditional networks.

The characteristic of error rates to be human-like is an often used goal. Out of all characteristics listed above, this one is the most addressed one in modern computer vision [Krizhevsky et al., 2012, Taigman et al., 2014, He et al., 2015, Russakovsky et al., 2015, Geirhos et al., 2017, Ho-Phuoc, 2018].

Lastly, the consideration of human-like errors, not only during inference but also during training, is much less prevalent than simply looking at the error rates. In a study proposed by [Geirhos et al., 2017], object recognition robustness is compared between humans and deep learning models. They conclude that "there are still marked differences in the way humans and current DNNs process object information. These differences, in our setting, cannot be overcome by training." Deep learning libraries like PyTorch and Tensorflow introduced the interpretability tools *Captum*

[Kokhlikyan et al., 2020a] and *tf-explain* [Meudec, 2021] respectively to make it easier to understand what the network is learning and where it breaks – a welcome step to open the black box of deep neural networks [Shwartz-Ziv and Tishby, 2017].

To the best of our knowledge, no approach exists that tries to test any of the characteristics as mentioned above for visual behaviours at a detailed level because there has been no detailed data until this dissertation. Chapter 3.3 shows the exact steps that are taken to solve the same-different task, the amount of data required, error rates and kind of errors, as well as evidence that humans seem to not learn this task directly.

With the availability of our data, we compare our findings with dominant current methodologies in machine learning. We have provided a brief overview in Section 5.1.2 – all methodologies, except for reinforcement learning, are mainly concerned about correct input-output behaviour. Reinforcement learning with its inclusion of an *environment*, *agent* and *actions* seems most suited to learn the same-different task and explore the human-like characteristics described earlier, particularly to investigate performed actions. Other methods, falling into the realm of supervised and semi-supervised learning methods, do not seem to fit our task due to the requirement of a large data set, labelled data and no component to actively control the input data in their standard interpretation. Next, we will provide a brief introduction to reinforcement learning.

5.2 A Foundational Introduction to Reinforcement Learning

Reinforcement learning currently enjoys popularity from the research community, especially with the recent successes of deep learning and the availability of rich datasets and better simulators.

For instance, the well-studied *Atari 2600* games provide a testing environment for reinforcement learning agents with dozens of classic Atari games [Bellemare et al., 2013]. Current, best-in-class reinforcement learning approaches are able to achieve human-level performance at about 40 out of the 57 games, and in some games, even above-human performance [Hessel et al., 2018]. In order to accomplish this, the approach coined *Rainbow* combines different improvements in deep reinforcement learning, such as duelling networks, multi-step learning, noisy nets, and others.

Work coming from Google's DeepMind and others challenge the best players in board games like Chess, Go and Shogi, and video games like Starcraft II, Minecraft and Dota 2, often reaching

or even surpassing levels of the best human players. [Vinyals et al., 2019] proposes a system called AlphaStar that achieves grandmaster level[1] in StarCraft II using multi-agent reinforcement learning in combination with imitation learning. Firstly, the system imitated behaviours from a pre-recorded dataset of 200 years worth of playtime. Secondly, the system then trained on a population of agents learning from many thousands of parallel instances of StarCraft II. Figure 5.7 shows the game interface of StarCraft II while AlphaStar is playing against one of the best human players in the world, *LiquidTLO*.

Figure 5.7: Screenshot of AlphaStar playing StarCraft II. Source: `https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii` (accessed: Feb. 24, 2022)

Real-World applications can be found as well, but are much more scarce. For instance, [Levine et al., 2018] used 6-14 robotic manipulators with different camera placement and hardware to train a system to learn Hand-Eye coordination for robotic grasping over the course of two months. In another example, [Manderson et al., 2020] proposes a vision-based method for unmanned underwater vehicle navigation.

Reinforcement learning is described as a *computational* approach to learning from interaction

[1]StarCraft II defines a Grandmaster as a player that is better than 99.6% of officially ranked human players. In fact, AlphaStar was rated at Grandmaster above 99.8% ranked human players.

by [Sutton and Barto, 2018]. That we learn by interacting with our environment seems very natural to us when we think about the nature of learning. For instance, when an infant plays, it has a direct sensorimotor connection to its environment. The more the infant plays and therefore exercises this connection, information about cause and effect, about the consequences of actions, and what to do in order to achieve goals is collected.

Hence, it can be defined as "Reinforcement learning is learning what to do – how to map situations to actions – so as to maximize a numerical reward signal [Sutton and Barto, 2018]." In comparison to other learning methods, such as supervised and unsupervised learning, here, the learner is not told which action to take, instead has to discover which actions yield the most reward by trying them. This also includes cases in which the action affects not only the immediate reward but also the next state and, therefore, all subsequent rewards. In [Sutton and Barto, 2018], the *trial-and-error search* and *delayed reward* characteristic, are the two main distinguishing characteristics of reinforcement learning.

5.2.1 Elements of Reinforcement Learning

A reinforcement learning system defines a problem using various, well-defined elements; *agent*, *environment*, *policy*, *reward signal*, *value function*, and, optionally, a *model* of the environment.

- *Agent* – The learner or decision maker is called the *Agent* and can be described as an entity that perceives, explores and acts on the environment.
- *Environment* – The *Environment* stands for the situation in which the agent exists. The *Agent* is not instructed about the environment, but rather, from the view of the *Agent*, the *Environment* is defined by an action and observation space. The action space defines the action(s) an agent can execute, while the observation space describes what the *agent* can sense/perceive. Both spaces can be either discrete or continuous and are limited by defined boundaries.
- *Policy* – The *Policy* describes the mapping from perceived states of the environment to actions to be taken when in those states. In other words, it defines the way the agent behaves at a given time. The *Policy* is crucial to the *Agent* as it is alone sufficient to determine

behaviour. *Policies* may be stochastic, specifying probabilities for each action. Furthermore, reinforcement learning algorithms can be divided into using *on-policy* and *off-policy* methods. More information will be provided in Section 5.2.3.

- *Reward Signal* – The *Reward Signal* defines the goal of a reinforcement learning problem. At each time step, the environment sends a single number called the reward to the *Agent*. The sole objective of the *Agent* is to maximize the total reward accumulated over the long run. In other words, the *Reward Signal* defines what are good and bad actions for the *Agent*. It is also the main basis for altering the policy; if an action selected by the policy is followed by a low reward, then the policy may be changed to select another action in that situation in the future.

- *Value Function* – The *Value Function*, similarly to the *Reward Signal*, informs the *Agent* how well or badly it is doing. The key difference, however, is that the *Value Function* specifies what is good in the long run, whereas the *Reward Signal* indicates what is good in an immediate sense. For instance, a state might always result in a low reward but still have a high value, as it is regularly followed by other states that yield high rewards.

- *Model* – The *Model* of the *Environment* mimics the behaviour of the environment and allows inferences about how the *Environment* will behave. This element, as opposed to the others, is optional. However, if a reinforcement learning system includes a *Model* it is called a *model-based* method, instead of a *model-free* method.

5.2.2 Markov Decision Processes

This section has been adapted from [Sutton and Barto, 2018] in order to provide the necessary background of this work.

Markov decision processes, or MDPs, are a classical formalization of sequential decision making, in which actions do not only influence the directly received reward but also subsequent situations, or states, and through those future rewards. Therefore MDPs involve delayed reward and the need to tradeoff immediate and delayed reward. MDPs are a mathematically idealized form of the problem of learning interaction to achieve a goal.

The agent and environment interact continually in the way that the agent executes a specific action, and the environment responds with a new situation to the agent.

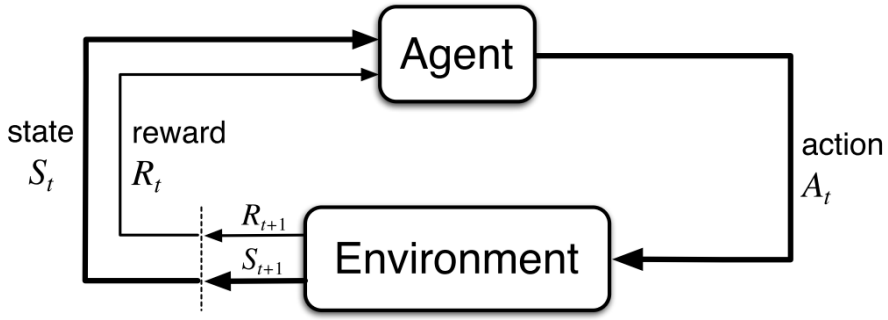


Figure 5.8: The agent-environment interaction in a Markov decision process. Source: [Sutton and Barto, 2018]

More specifically, the agent and environment interact at a sequence of discrete time steps, $t = 0, 1, 2, 3, \ldots$. At each time step, t, the agent receives a representation of the environment which is called the state, $s_t \in S$. The agent, in return, selects an action, $a_t \in A(s)$. For each executed action, a_t, the environment returns a new state, s_{t+1}, and a numerical reward, $r_{t+1} \in R \subset \mathbb{R}$. Figure 5.8 provides an illustration of this interaction of agent and environment.

The reinforcement learning problem can be framed as a MDP by defining the set of states, actions, and rewards:

$$(S, A, R). \tag{5.1}$$

The MDP and agent thereby produce a sequence that begins as:

$$s_0, a_0, r_1, s_1, a_1, r_2, \ldots \tag{5.2}$$

In a *finite* MDP, this sequence has a finite number of elements and the random variables r_t and s_t have well-defined discrete probability distributions dependent only on the preceding state and action. That is, the probability of each possible value for $s_t + 1$ and $r_t + 1$ depends only on the present state s_t and a_t, and not at all on earlier states and actions. The assumption that the future state is independent of the past given the present is called the *Markov Property* and is mathematically formulated as:

$$\mathbb{P}[s_{t+1}|s_t] = \mathbb{P}[s_{t+1}|s_0, s_1, s_2, \ldots, s_t]. \tag{5.3}$$

The transition from state, s_t, to the next state, s_{t+1}, by executing the action, a_t, and receiving the reward, r_t, is called a *transition step* and a sequence of such steps is called a *Markov Chain*. The transition step is expressed in the *state-transition probability function* P with probability $\mathbb{P}$:

$$P(s', r|s, a) = \mathbb{P}[s_{t+1} = s', r_{t+1} = r|s_t = s, a_t = a], \tag{5.4}$$

for all $s', s \in S$, $r \in R$, and $a \in A(s)$. From this function P, we are can compute many things we want to know about the environment, such as the *state-transition probabilities*:

$$P^a_{ss'} = \mathbb{P}[s_{t+1} = s'|s_t = s, a_t = a] = \sum_{r=\mathbb{R}} P(s_{t+1}, r_t|s_t, a_t). \tag{5.5}$$

It is also possible to compute the expected rewards for state-action pairs:

$$R(s, a) = \mathbb{E}[r_{t+1}|s_t = s, a_t = a] = \sum_{r=\mathbb{R}} r \sum_{s'=S} P(s_{t+1}, r_t|s_t, a_t). \tag{5.6}$$

The default objective of a reinforcement learning system is to maximize the cumulative reward. Depending on the type of reinforcement learning algorithm, the goal is to either learn a policy, a value function or a model that maximizes that objective.

In the described situation so far, the MDP is fully observable, which means that we know all possible actions and states. However, for some problems, this cannot be guaranteed. If the environment is uncertain and we do not know all actions and states, it is called a *partially observable MDP* (*POMDP*). In this case, the gathered information about previously visited states is saved in memory and used to make decisions [Sutton and Barto, 2018].

The MDP framework is abstract and flexible and can be applied to different problems. Problems can be low-level controls, such as turning on or off the light, or high-level controls, such as forecasting the weather.

Many different reinforcement learning systems have been developed over the years. It is beyond the scope of this document to go into further detail, but the interested reader is referred to [Sutton and Barto, 2018, Kaelbling, Leslie Pack and Littman, Michael L and Moore, 1996, Arulkumaran

et al., 2017] for a deeper investigation.

Continuing, we give a brief introduction of the two reinforcement learning systems that we have chosen to learn the Same-Different task, PPO and SAC, and explain why we chose them.

5.2.3 Policy

In this work, the two reinforcement learning algorithms used are a *on-* and *off-policy* method.

Given a state, s_t, the policy, π, which is either an algorithm, a function, or a set of rules, describes which action, a_t, will be taken by the agent:

$$a_t = \pi(s_t) \tag{5.7}$$

The policy chooses the action that maximizes the cumulative reward from a given state, s_t, hence optimizing the long-term reward instead of the immediate reward. π can technically be anything as long as it takes a state and returns an action. Further, a policy can be classified into two types based on its return: *deterministic policy* if a single deterministic action is returned or *stochastic policy* if a probability for each action is returned. The latter is denoted as:

$$\pi(a_t|s_t) = P_\pi[a_t|s_t], \tag{5.8}$$

where $\pi(a_t|s_t)$ is the normalized probability vector of all actions. This is useful as it takes into account the dynamics of the environment, hence helps its exploration. Both algorithms used for this experiment use a *stochastic policy*.

Another differentiation of reinforcement learning algorithms is how the policy is improved during the learning process. Here, two methods are used; *on-policy* and *off-policy*. On-policy means that the total future return is estimated assuming the current policy continues to be followed. Off-policy, on the other hand, assumes a greedy policy[2] is followed despite the fact it is not following a greedy policy.

[2]A greedy policy takes the action that is believed to yield the highest expected reward [Sutton and Barto, 2018].

5.3 Same-Different & Reinforcement Learning

In this section we learn the Same-Different Task using reinforcement learning. It is divided into four parts; Section 5.3.1 explains the environment and algorithms used to train the reinforcement learning agent, Section 5.3.2 presents the results, Section 5.4 analyzes the outcome, and lastly Section 5.5 provides a summary.

5.3.1 Setup

Here, we present the setup of our reinforcement learning System to approach the Same-Different task. This includes a simulated environment (Figure 5.9), as well as our selected algorithms used to train the Agent.

The central element of most reinforcement learning systems is the environment with which the agent interacts (See Figure 5.8). The environment can be either implemented in the real world or a simulated one (hybrid methods exist too). While a robotic agent that trains in the real world benefits from the realism the world provides, for instance, realistic sensor readings, including sensor noise, errors, slippage and others, the real world does not run faster than in real-time. Reinforcement learning requires many, often more than hundred of thousands, iterations to converge. Running in real-time presents, therefore, a critical bottleneck. However, if the available space and the budget permit, multiple instances can be run in parallel to reduce the training time. For example, [Levine et al., 2018] presents a system to learn hand-eye coordination for robotic grasping in the real world. In their setup, their "large-scale data collection setup consists of 14 robotic manipulators" which were run over the course of two months consecutively.

We do not have the resources to implement multiple *PESAO* environments and equip each of them with a mobile robotic platform. [Levine et al., 2018] collected over 800,000 grasp attempts to learn the task. The *Three-Dimensional Same-Different Task for Active Observers* required on average 47.52 seconds per attempt which brings us to 38,016,000 seconds or almost 15 months for 800,000 attempts. This assumes that the robotic platform moves as fast as a human, runs 24/7 without any interruptions (no hardware failure, no charging, ...) and changing the objects between trials does not take any time. It is safe to say that this would take well beyond 15 months. Therefore, we use a simulator that not only allows us to run beyond real-time and in parallel

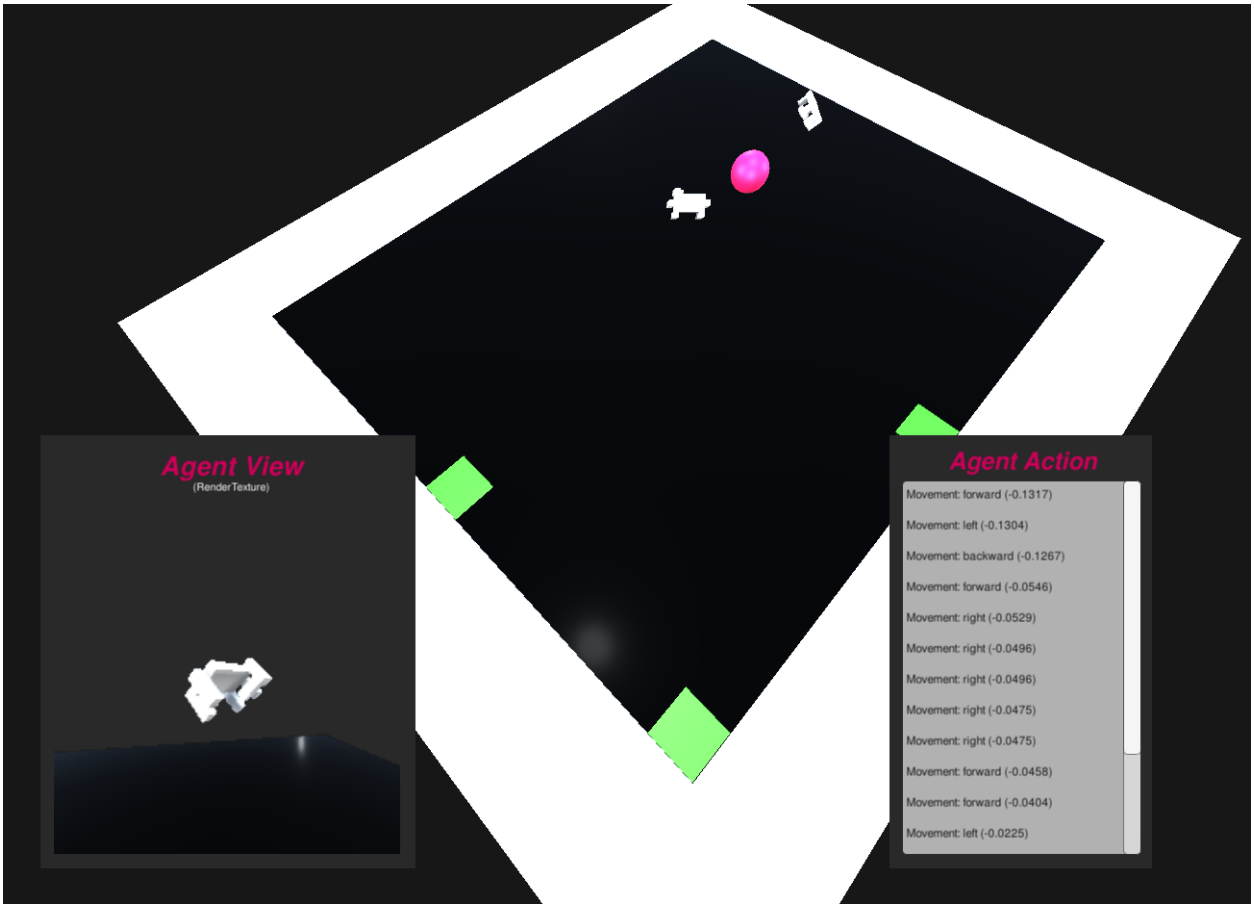


Figure 5.9: The virtual Three-Dimensional Same-Different environment we used to evaluate reinforcement learning methods. We have implemented the environment using Unity (`https://unity.com`) and its machine learning extension ML-Agents [Juliani et al., 2018]. In the background, the environment can be seen with its boundaries (white border) with which the agent (purple sphere) interacts. The green squares are different starting positions (not observable with the agent's sensors), and the stimuli are in the center of the environment. In the foreground on the left, the render texture is shown to illustrate what the agent is observing (Agent View). On the right, the list of actions performed by the agent is recorded, including the accumulated reward, for instance, "Movement: forward (-0.1317)." The dimensions for this environment are taken from the original experimental setup (see Chapter 2.3 for details) as used by our human subjects.

virtually for free but also allows us to explore different versions of the environment easily.

Many different frameworks and libraries exist to create simulated reinforcement learning environments. Mostly, they are tailored towards a certain task (e.g. Atari Games [Bellemare et al., 2013], Autonomous Vehicles [Shah et al., 2018]) or a specialty (e.g. physical realism [Coumans and Bai, 2016], photo-realism [Juliani et al., 2018], etc.). So, it can be a non-trivial task to find the right framework. For the Same-Different environment we are looking for a framework with characteristics that allows for:

1. the customizability to re-create the same-different task as presented in Section 2.1.2.

2. the graphical fidelity to sense the environment in a similar quality as the real environment.

3. complex sensors, especially visual sensors.

4. fast simulation.

5. fast prototyping.

While bulletpoints 1-3 are somewhat self-explanatory as we are trying to bring as much of the characteristics of the real task to a simulation, we want to provide some clarification for points 4-5.

Fast simulation, prototyping, often together with distributed computation, are requirements not of the task per se, but attributed to the learning method. Reinforcement learning, similarly to other machine learning methods, such as supervised and unsupervised methods, needs data and computational resources to learn the task. [Jordan and Mitchell, 2015] calls these methods "...data-intensive machine-learning methods... ." While reinforcement learning does not necessarily need a curated data set, it rather creates its own data by using observations, rewards depending on the sample efficiency of the algorithm; often, billion of samples are needed to converge to an optimal solution [Badia et al., 2020, Espeholt et al., 2018]. The sheer complexity of visual data does not help here either [Tsotsos, 1987]. This is why reinforcement learning researchers have addressed some of these issues by improving simulators that can run at high speeds (e.g. $\approx 400,000$ dynamics evaluations per second [Todorov et al., 2012]), allow for distributed and parallel execution [Juliani et al., 2018], and fast prototyping [Juliani et al., 2018].

Based on all of these criteria, we chose Unity with its ML-Agents framework [Juliani et al., 2018] to simulate our environment. While it does not provide best-in-class simulation speeds, it checks all other characteristics we are looking for. For further information, including comparisons to other simulators, please see [Juliani et al., 2018].

The environment we have implemented simulates the original, real environment of the three-dimensional same-different task. Figure 5.9 shows the graphical user-interface implemented, including debugging features. Figure 5.10 provides annotations about the different elements of the user-interface, which we will describe now.

Figure 5.10: The same environment as shown in Figure 5.9 but annotated with descriptions of the main user interface elements.

- *Tracking Area* – As shown in Figure 2.39, the subject, here agent, can move freely around within an area of $430cm \times 340cm$. The agent cannot move beyond these limits.

- *Objects* – The objects used are the CAD models of the L_2 *TEOS* set. The objects are scaled appropriately to match the size of the real-world objects. Furthermore, their orientation is determined by the same script used for the real-world experiment.

- *Starting Position* – Three starting positions are defined, corresponding to the real-world experiment. While the user interface shows green patches for each position, the agent does not see these patches. As with the object orientations, the starting position is randomly determined with the same script used for the real-world experiment.

- *Agent* – While the agent is visualized in the user interface as a purple sphere, technically, it is realized as a floating scene camera. The agent's movements are restricted to the boundaries

of the Tracking Area, 50cm - 220cm in height, and by collision bodies surrounding each object to avoid the agent to "go through" the objects.

- *Observation* – The observation is realized using a grey-scale render texture with a resolution of 200×200 pixels. The camera rendering to the texture moves with the agent and has a field of view of $90°$ and perspective projection.

- *Action Taken* – This panel records all actions that have been taken in the current episode. Besides logging the actions (e.g. *Movement: forward*), the panel also shows the current, cumulative reward (e.g. *-0.1317*).

In terms of the action space, the agent is implemented with twelve actions: forward, backward, left, right, up, down, yaw increase, yaw decrease, pan increase, pan decrease, answer *same*, and answer *different*. All actions are implemented as discrete actions moving the agent 10cm or $10°$, respectively if moving or turning, each time an action is executed. Providing an answer automatically terminates the episode.

The observation space is a 200×200 pixels grey-scale image at each step of the simulation. The image is rendered from the agent's position and orientation[3].

For this environment, we have used a sparse reward which was only issued at the end of an episode (when the agent casts an answer). In contrast, dense rewards can be used to guide the agent towards the desired goal [Sutton and Barto, 2018]. Since we want to see *what* a modern machine learning approach would learn and *how* it would solve this task, in order to compare it to the findings of Section 3.4, we used a sparse reward. It is simply defined as

$$R = \begin{cases} +1, & \text{if } answer = True \\ -1, & \text{otherwise.} \end{cases} \tag{5.9}$$

If a correct answer is provided, the episode will be terminated with a $+1$ cumulative reward, and if the answer is incorrect, the episode will terminate with $R = -1$. Other reward functions have been tried as well, including a dense reward function. For instance, the agent receives a

[3]A sphere does not have a natural orientation. However, in our case, the camera is pointing in $z - direction$ with its $up - vector$ alligned with the $y - axis$ of the environment at time-step $t = 0$. For $t > 0$, the camera moves together with the sphere.

reward after each action – positive if the agent is looking at one of the objects, negative otherwise, discounted over time[4], and accompanied with another positive or negative reward for a correct or incorrect answer, respectively. Many variations of reward functions have been tested (about a dozen); however, the sparse reward function, as shown in Equation 5.9 performed best.

To train the agent, we have chosen two reinforcement learning algorithms; Soft Actor-Critic (SAC) [Haarnoja et al., 2018] and Proximal Policy Optimization Algorithms (PPO) [Schulman et al., 2017]. Both algorithms are widely used and have shown wide adoptions to different tasks. The main difference between these two algorithms is that PPO is an on-policy approach learning directly from the data, whereas SAC is an off-policy approach learning from a buffer of stored data from past episodes. For further details on each algorithm please see [Haarnoja et al., 2018] (SAC) and [Schulman et al., 2017] (PPO). Hyperparameter are empirically set as specified in Table 5.1.

Hyperparameter	**SAC**	**PPO**
Network	IMPALA Resnet [Espeholt et al., 2018]	IMPALA Resnet
Layers	3	3
Hidden Units	512	256
Batch Size	128	128
Buffer Size	2048	2048
Learning Rate	3.0e-4	3.0e-4
Max Steps	10^8	10^8

Table 5.1: SAC and PPO Hyperparameter using Unity's ML-Agents.

Where "Network" specifies the neural network used to approximate the policy π, "Layers" is the number of hidden layers used for the neural network counting from after the CNN encoding of the visual observation,"Hidden Units" corresponds to the number of units in each fully connected layer of the neural network, "Batch Size" describes the number of experiences used for one iteration of a policy update, "Buffer Size" stands for the number of rewards obtained before the model is updated, "Learning Rate" is the strength of each update step, and "Max Steps" corresponds the steps of the simulation during the entire training process.

As Section 5.3.2 will show, neither of the reinforcement learning algorithms was able to learn anything useful. To understand why, we have simplified the task in many regards to make it

[4]Discounted if the number of observations exceeded the average amount of observations of humans (92.38). Once this number was reached, in a step-wise function, the reward was minimized for looking at objects until it became negative.

"easier". In the current set up the agent needs to learn multiple key behaviours before learning the concept of same-different, for instance:

- How to move
- Observing the objects
- Know "what" to look for
- Know "when" to answer

Let us take for instance "Observing the objects" and perform a back-of-the-envelope calculation to see how complex this task is alone. The environment is $430 \times 340cm$ in size, the agent can move between $50 - 220cm$ in height ($170cm$ range of motion). Each movement covers $10cm$. To observe an object, the agent has

$$\frac{430}{10} \times \frac{340}{10} \times \frac{170}{10} = 24,854 \tag{5.10}$$

positions to do that. Considering that this task involves two objects, this number doubles to $49,708$. Now, we have to keep in mind that we start with a vanilla agent, actually looking at an object needs to be learned as well. For each of the almost $50,000$ positions, the agent can choose a "look at" direction. The agent can pan $360°$ and tilt $180°$ to cover the entire viewing sphere around it. This adds another

$$\frac{360}{10} \times \frac{180}{10} = 648 \tag{5.11}$$

possibilities for each position and brings it to

$$648 \times 49,708 = 32,210,784 \approx 3.22 \times 10^7. \tag{5.12}$$

Considering the large action sequence presented in Section 3.4, while having shown that humans are very good at this task, the remaining key behaviours are likely to add further complexity to the task. Therefore, we have decreased the complexity of the task in three ways; object complexity, action space, and observation space. Table 5.2 gives a brief overview of all the environments

implemented and their characteristics. Note: For each action space, the agent has two additional actions to cast the answer (*same, different*).

Figure 5.11 presents the RL same-different environments implemented. In total, four different environments have been used to systematically simplify the learning task and to investigate where the reinforcement learning methods will break.

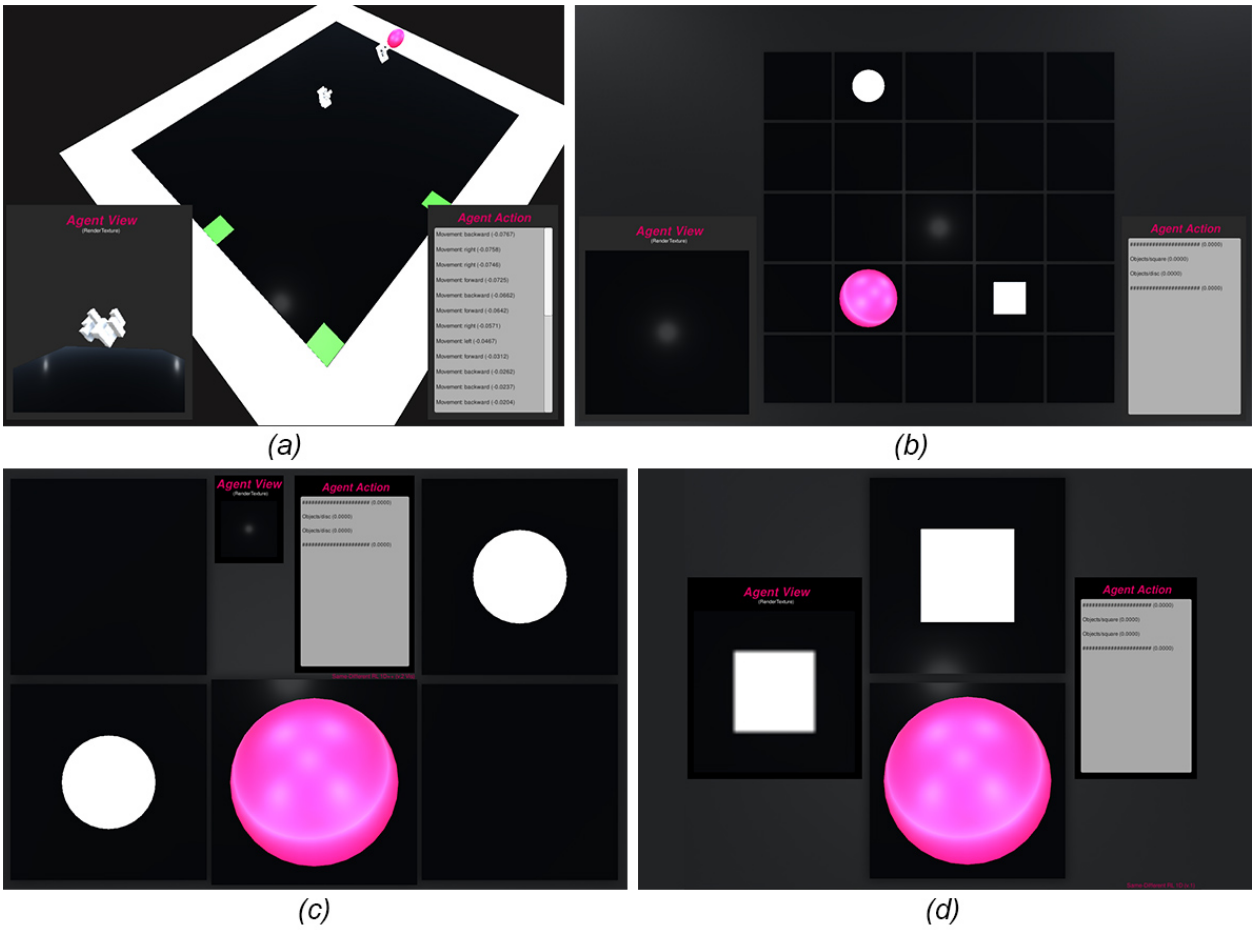


Figure 5.11: Different instantiations of the RL Same-Different Environment in order to simplify the task. The magenta sphere illustrates the agent in all instantiations. ***(a)*** shows the full 3D environment as used in the instantiations I-IV (green squares stand for the different starting positions), ***(b)*** shows the simplified environment to a 2D grid (instantiation V) with the stimuli also simplified to a white square and circle, ***(c)*** shows the simplified environment to a "1D grid" (instantiation VI), and ***(d)*** shows the simplest environment which does not involve any "search" as each action will show one of the two objects (instantiation VII).

For the environment instantiations V-VII we also updated the hyperparameters. Table 5.3 shows the hyperparameters used. In comparison to the hyperparameters to train on the more complex environment (Table 5.1), a much smaller CNN is used to learn the policy. This CNN

Environment	Action Space	Observation Space	Object Complexity
I	Full 6 DOF with $10cm$ and 10° partitioning as described before.	$200px \times 200px$ grey-scale	Twelve 3D Objects in three different complexity levels (*TEOS* L_2)
II	Same as environment I, but the agent automatically looks at one of the two selected objects	*As before*	*As before*
III	Environment subdivided into 700 Voxels. Actions $1, 2, 3, \ldots, 700$ moves the agent to the corresponding voxel. The "look at" is automatically set like in II.	*As before*	*As before*
IV	*As before*	*As before*	Objects are simple geometric shapes: sphere and cube. See Figure 5.11 b), c), and d)
V	Agent moves on a 2D Grid (5×5) in four directions (*up, down, left, right*). Looking from above at a cell. An object (if present) fills the cell. (This makes it a two-dimensional version of this task.)	*As before*	*As before*
VI	Similar to before, but the Agent has two 1×2 fields (for each object) to search for the objects. The action space simplifies to (*change_object_1, change_object_2*).	*As before*	*As before*
VII	Similar to before, but the Agent moves between both objects back and fourth with the same action (*change_object*); hence, no "search" involved.	*As before*	*As before*

Table 5.2: Environment instantiations based on three different parameters.

is only made of two convolutional layers with 128 hidden units. The reason behind this is that the visual input has less variation; hence a less complex network is needed. In contrast, a larger network, as it was tried as well, will not learn anything.

Hyperparameter	**SAC**	**PPO**
Network	Simple CNN	Simple CNN
Layers	2	2
Hidden Units	128	128
Batch Size	128	128
Buffer Size	2048	2048
Learning Rate	3.0e-4	3.0e-4
Max Steps	10^8	10^8

Table 5.3: SAC and PPO Hyperparameter using Unity's ML-Agents for environments V-VII.

Lastly, for training, we have used three different machines. Equipped with NVIDIA graphic cards (Titan Xp, 1080 Ti, Titan Z) and AMD CPUs (AMD Threadripper 2990WX, 1950X, and Ryzen 7 5800X). To accelerate the training we have trained using four concurrent Unity instances.

5.3.2 Results

In this section, we will present the results of training seven different versions of the same-different task using SAC and PPO. In order to do so, we will show the learning progress of the cumulative reward for the best training scenario ($algorithm \times environment$) and also present a higher-level analysis, including the average amount of movements and overall performance.

Figure 5.12 presents the training progress on all seven environments. The plot shows the cumulative reward against the training step. The range of the $y - axis$ goes from $[-1; 1]$ which is the return of the reward function (see Equation 5.9)[5] Each environment was trained for 100 million steps following our empirical investigations and recommended best practices by [Lapan, 2018, Ravichandiran, 2020].

To understand the cumulative reward, we need to take into count how the graph was generated. Every 2000 steps, the average cumulative reward was stored and plotted. A cumulative reward of -1 means that the agent consistently provided an incorrect answer for 2000 steps, while a reward of 1 means that the answer is correct all the time.

[5]We have investigated other reward functions as well. However, this function worked best for this scenario.

Figure 5.12: Training Progress on Environments I-VII. Plotted as cummulative reward vs. training step (0 - 10^8). Only the best performing algorithm of each environment is plotted (noted in the legend).

If we look at any of the three-dimensional environments (I-V), the cumulative reward never leaves 0.0 by much, and this means that the agent providing as many correct as incorrect answers. Technically, achieving a performance of 50%, however taking into account that this is a two-choice task, the agent could have just been guessing. In fact, a deeper look at the actions the agent chose, it was barely moving (on average 0.5 movements with a standard deviation of 0.7) and jumping right to the answer. In another example, if we look at environment VII, at about 85 million steps the cumulative reward jumps from about 0.75 to about 0.90 and slightly increases to 0.98. This is the easiest environment but also the only one that was able to be learned to this extent.

With a drop in performance to about 0.6, environment VI was learned using PPO best. The drop in performance can be explained with the increase in dimensionality of the environment. Now, the agent is not always presented with one of the two objects as it was the case for environment VII, but rather has to "look" for the objects which could be in either of two cells. Notably, the training progress for this environment started unstable for the first $\approx$ 18 million steps jumping from 0.0 down to -0.4, then up to about 0.20, to then plummet to -1.0 at 10 million steps. While the run stabilized well after, this can be caused by multiple factors such as a learning rate that

was set too high, a disadvantageous ratio of exploration vs. exploitation, batch size set too small. Reinforcement learning methods are notoriously unstable during training, hence requiring a lot of fine-tuning [Nikishin et al., 2018].

Another drop in performance to 0.31 is observed for environment V. Environment V adds another dimensionality to the action space, as here the agent moves on a two-dimensional grid, needing to decide in which of the four directions (*up, down, left, right*) to go. The agent has to find the two objects in two out of 25 cells. While the task seems harder than the one of environment VI, the performance, as we will discuss shortly, dropped only slightly.

Lastly, all three-dimensional environments (I-V) which is the same environment with variations in their action space (Table 5.2), were unable to be learned. We have tried dozens of different hyperparameter settings and reward signals, but none of them seem to allow either SAC or PPO to learn these environments. This also means that no direct comparison to the real-world three-dimensional Same-Different Task to what has been learned using reinforcement learning can be made, unfortunately.

Moving forward, we will exclude the three-dimensional environments (I-V) from the analysis. These agents have not learned any meaningful behaviours – not moving at all, not looking at the objects, wildly guessing the answer, and so on.

Figure 5.13 presents an accuracy (left) and movement (right) analysis for environments V-VII. All agents have been evaluated 1000 times in each environment to avoid sampling biases.

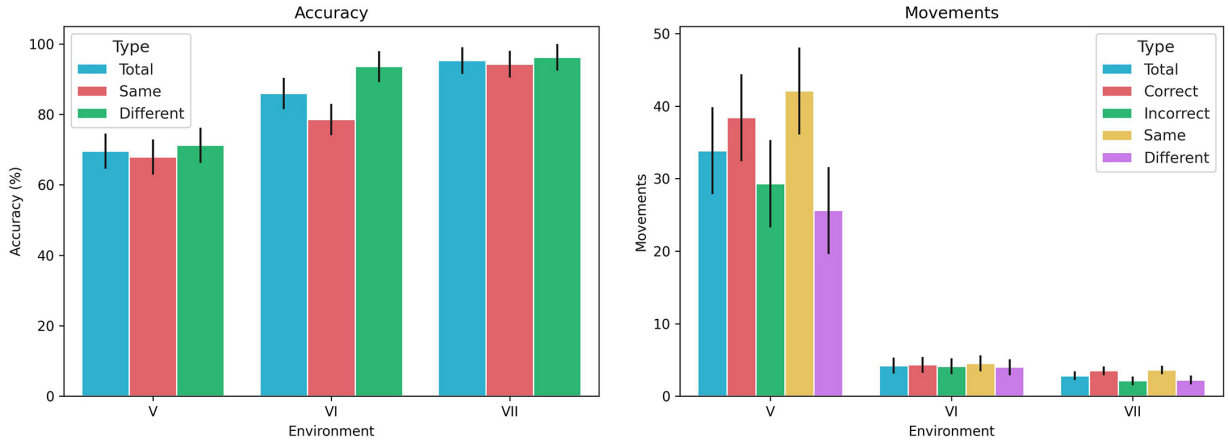


Figure 5.13: Learned performances on Environments V-VII. A high-level analysis of the accuracy (left) and number of movements (right) executed by each agent to solve the one- or two-dimensional version of this task.

We have plotted the average accuracy (blue), accuracy for cases in which both objects are the same (red) and objects are different (green). The agent trained on environment V achieved the lowest accuracy of 69.54%, trained on environment VI the accuracy increased to 85.94%, and for environment VII the accuracy is 95.25%. The drop of accuracy between environment V and VI correspond well with the increase in the size of the environment and increase of the action space. However, while environment VII is only slightly smaller (two vs. four cells), the accuracy only climbed by about 9% to 95.25%. In the grand scheme, this is an impressive accuracy, but such a simple task should have been learned perfectly. Especially after the objects are always present in the camera view, hence no "search" to find the objects is involved. In comparison, humans achieved an absolute mean accuracy of 93.82% throughout all evaluated combinations, and this task was in three-dimensions, with objects much more complex than these. Generally, the same object pairing was less accurately answered than the different one by the agent. This stands in contrast to what humans did; there the same case achieved a higher accuracy. This potentially leads to a significant difference in how this task is solved.

Similarly to the accuracy, the amount of movement (number of executed actions of the agent except the action to answer) corresponds with the complexity of the environment; V required 33.85 movements on average, VI 4.2 movements, and VII 2.8. For all environments the agent successfully learned how to traverse it. Environment V consists of a 5×5 grid, so in total 25 cells which the agent always visits at least once. Environment VI and VII have four and two cells, respectively and for both the same behaviour is observed. While it is not possible to observe both objects in environment VII with less than two movements, it is for environment V and VI. Two behaviours led to this number; either the agent revisited empty cells multiple times (not remembering it has already visited this cell) or the agent did not provide an answer after the second object was observed immediately. An example of both behaviours are shown in Figure 5.14.

Furthermore, while the same scenario was answered less accurately, the agent also needed more steps for this case in general. For all three environments, a same object pairing always required most movements. This is identical to the observation of the human experiment. However, looking at the amount of movement for correct vs. incorrect answers, the agent does the contrary to humans; it performs more movements for correct answers, whereas humans show more movements before providing a false answer.

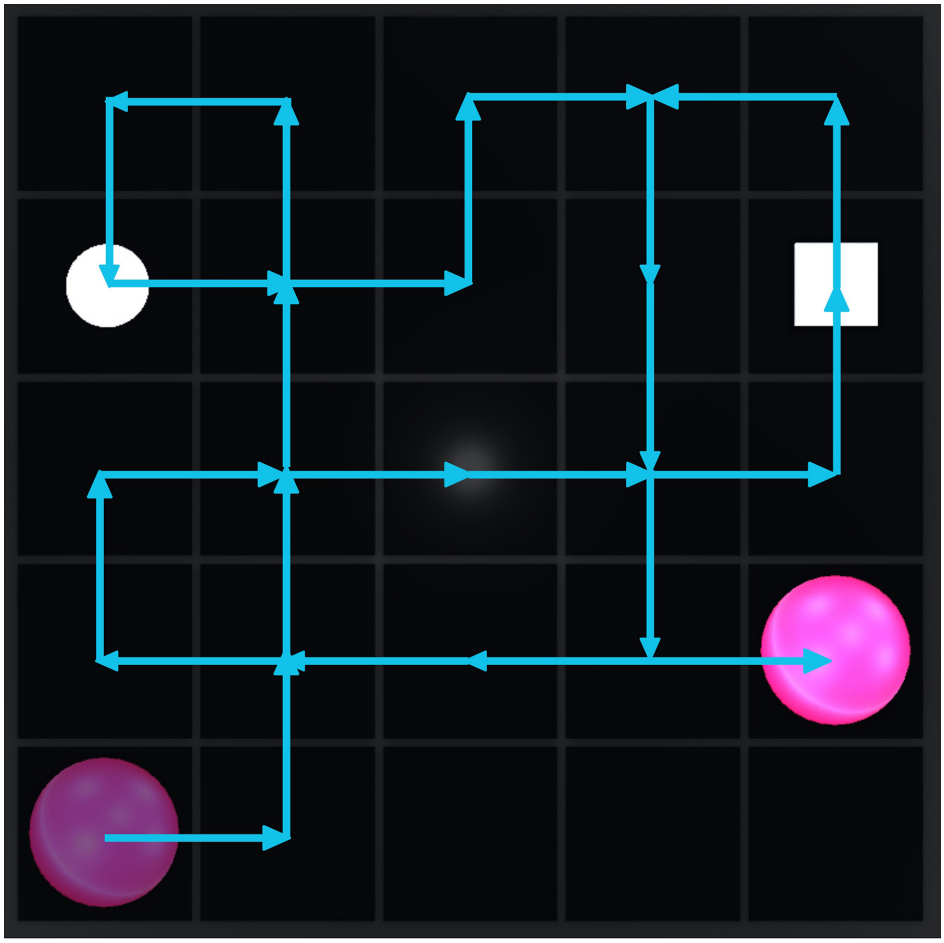


Figure 5.14: Example movement of the PPO agent (magenta sphere) in environment V. Each movement is illustrated as an arrow. In total 29 movements where executed before the episode ended with a correct answer. Notably, the agent performed six additional movements after observing the second object.

5.4 Interpretation of Results

As mentioned before, reinforcement learning has seen promising successes across different fields, the presentation of negative results in the previous section demands an explanation of why the same-different task was not successfully learned unless simplified. We will now explain why the methodologies underlying reinforcement learning do not match the same-different task.

5.4.1 Methodologies Do Not Match

The underlying assumption of reinforcement learning is that intelligence arises through trial and error, trying one thing, failing at it, trying something else until the goal is achieved.

However, implementing this in real life faces several challenges. If there are no constraints on the amount of data, computation and time, then we have the problem of an infinite number of monkeys typing for an infinite amount of time. The *infinite monkey theorem* describes that a monkey hitting a typewriter for an infinite amount of time will eventually type any given text, such as the complete works of William Shakespeare [Borel, 1913].

Trial and error only works if the domain is small enough and data exists to initialize with – both are not the case for the same-different task. For instance, AlphaStar was initialized with 200 years worth of StarCraft II playtime data. The lead of this project, Prof. David Silver, described this as a necessary step to overcome the exploration problem [Kelion, 2019]. Otherwise, discovering new strategies lacking any guide would be a "needle in a haystack problem" – with the agent required to stumble upon a series of steps with beneficial outcomes. The availability of mostly toy-like environments, such as the Atari 2600 games [Bellemare et al., 2013], VizDoom [Kempka et al., 2016], OpenAI Gym [Brockman et al., 2016] (Figure 5.15 shows images of some environments), OpenSpiel [Lanctot et al., 2019], and others highlight that reinforcement learning, as of now, is mainly applicable to smaller domains, with a limited observation and action space. This is also true for AlphaStar; the algorithm observed the game only through the overview map (Figure 5.7 bottom left) and a list of units – The majority of the visual information, as used by the human counterparts, was ignored.

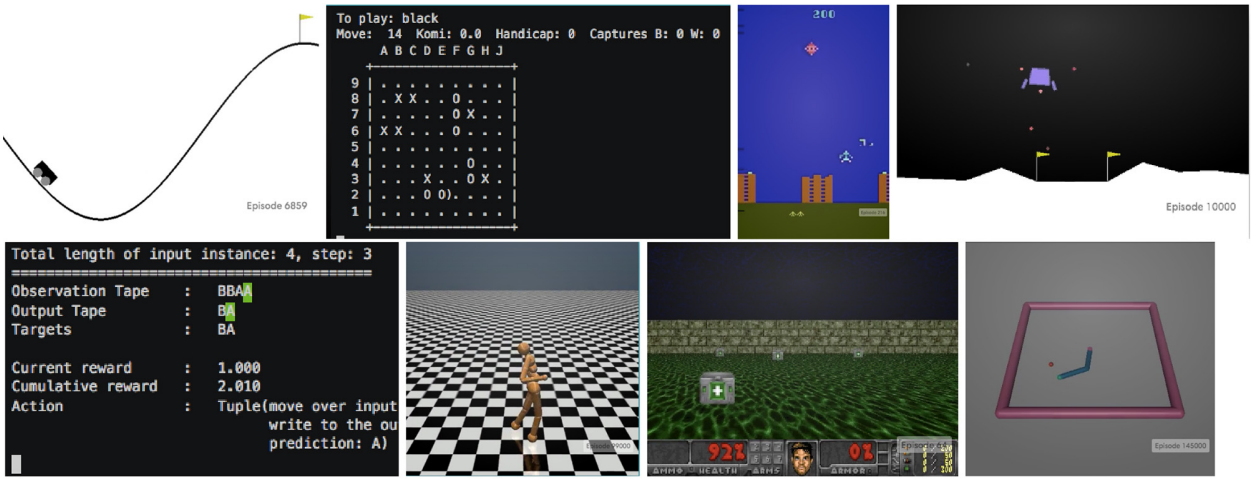


Figure 5.15: Images of some environments that are currently part of OpenAI Gym. Source: [Brockman et al., 2016]

In the reinforcement learning literature, an often given source of inspiration for trial and error

learning is human learning and the development of the human brain. [Sutton and Barto, 2018] goes so far to say that of all forms of machine learning, reinforcement learning is the closest to the kind of learning that humans and animals do. Specifically, the concept of reward-based learning, is central in reinforcement learning is analogously compared to the concept of pleasure (high reward) and pain (low/negative reward) of humans. [Silver et al., 2021] compares the main underlying concepts of reinforcement learning of reward maximization and trial and error to the initial configuration of natural intelligence, such as a human baby brain, and development of sophisticated abilities, such as a human adult brain, respectively.

This inspiration for machine learning methods, in general, is often due to Turing:

> "Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are, from our point of view, almost synonymous.) Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child." – [Turing, 1950a]

In the 1940s, this perspective was perfectly acceptable, considering the understanding of neuroscience and cognitive psychology. However, in over 70 years, much has changed. The blank sheet and the writing mechanisms change while learning and maturing. Evidence can be found in [Siu and Murphy, 2018], which summarizes the visual development milestones.

Figure 5.16 shows age across the top and milestones of visual function across the vertical axis. For each milestone, a green arrow means that it is still maturing, black means it has matured, and red means that it is starting to degrade. It can be noticed that it takes time for some of these to mature. For instance, face recognition takes until the age of 20 before it is fully developed, contrast sensitivity is not fully developed until the age of eight years, motion perception matures at the age of twelve years.

One might say that data is required to learn these milestones of that period of time. Certainly,

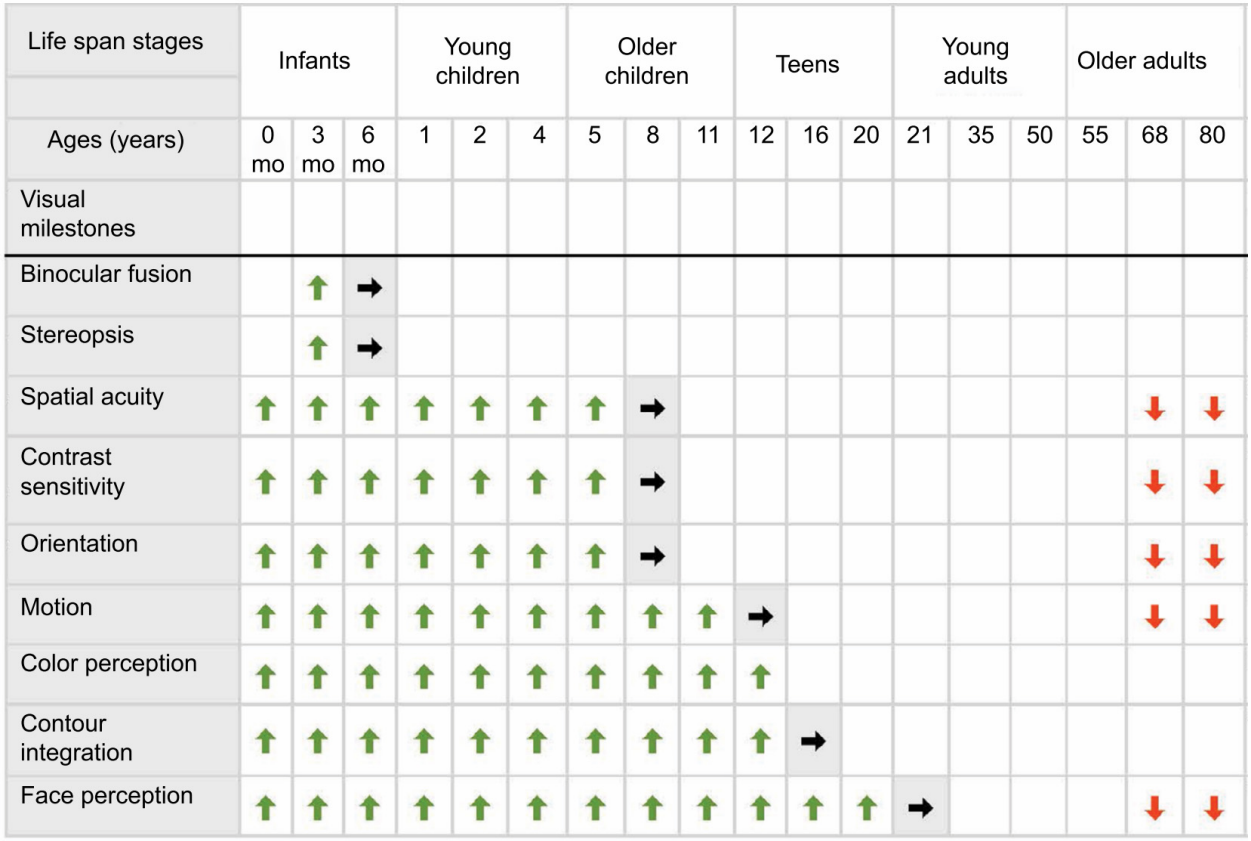


Figure 5.16: Summary chart for development of human visual milestones. A green arrow indicates that the visual milestone is developing. A black arrow means that the milestone has matured. A red arrow shows that the milestone is declining. Source: [Siu and Murphy, 2018]

data is required. However, [Siu and Murphy, 2018] also looked at the anatomical milestones of V1 and various aspects of anatomy and neurophysiology change over time as well. So, human learning is a much more complex activity than the way Turing described it. Not only are we using data, but the structure on which the data is being imprinted and learns the data matures over time while learning.

In conclusion, assuming that learning through trial and error is enough seems highly unlikely. In fact, looking at the results presented in Section 3.4, specifically Figure 4.2, humans do not approach this using trial and error. Remember, subjects have not seen the objects before, yet the results highlight that the problem was approached directed and not in a trial and error manner. Further, Subjects did not get better at this task throughout trials, again not a sign for trial and error learning. Perhaps this is the key to solving this task and explains why reinforcement learning does not learn it unless it is simplified.

Another aspect to consider is the underlying assumption of reinforcement learning that the future state is independent of the past given the present – the *Markov Property*. While the definition of state is per se flexible in the definition of a reinforcement learning problem, the concept of what a state is given a problem needs to be known during training. As for the same-different task, and as shown in Figure 4.2, subjects produce long sequences of often dependent actions (*Global Gist*, *Divide and Conquer*, *Outlier Detection*, *Coarse to Fine*, *Alternating Fixation/View*). The dependencies and hence the history of *what* has been observed *how* is important.

For instance, let us take the higher-level operation *Outlier Detection*. For the case of a different object pairing, similarly to the same objects, it is necessary to keep track of all potential outliers and test them one by one. Considered outliers can go as far back as the entire action sequence, which requires defining the current state as such. Strategies are deployed dynamically; hence the definition of the state and what it entails needs to be dynamic. Similar to what has been described earlier: the structure on which the data is being imprinted and learns the data matures over time while learning.

5.4.2 Suggestions

To summarize, the implications of this study for reinforcement learning are as follows:

1. Reward alone seems not enough, otherwise, discovering new strategies lacking any guide would be a "needle in a haystack problem "– with the agent required to stumble upon a series of steps with beneficial outcomes. As presented in 3.3, humans perform complex visuospatial behaviours in long sequences to solve a same-different task.

2. Trial and error for this task appears to be not enough for artificial intelligence unless the domain is small enough and sufficient data exists to imitate intelligent behaviour – both are not true for the same-different task.

3. Generally, the inspiration of human learning, based on Turing, is outdated – we cannot just use data, but the structure on which the data is being imprinted and learns the data matures over time while learning. The structure itself might have been learned over the course of evolution.

4. The underlying assumption of the classic *Markov Property* does not encompass what is needed to solve the same-different task – actions are dependent; hence the history of *what* has been observed *how* is essential.

Further, another differentiating factor between reinforcement learning and how humans learn this task is that humans do not learn it (Section 3.4); they use what they know to plan solution strategies. Hence, learning at the level of abstraction that reinforcement learning requires is not the right approach for this task.

In conclusion, reinforcement learning is an excellent method to solve a wide range of problems – its current popularity is understandable. It is already playing a big part in the development of artificially intelligent systems. However, reinforcement learning does not match the requirements of the same-different task. What is missing is sketched out in this section and hopefully helps to develop new methods and additions/modifications to existing methods. One of which is perhaps a progression of learning sub-tasks, fine and finer divided, that are then used by a planning system to solve new tasks. The sub-tasks correspond to small observable units of behaviour which makes them comparable to observed human behaviour, hence making them *human-like*. We call this *Progressive Learning* and point to a future research direction.

5.5 Summary

In this chapter, we revisited the original goal of AI as presented for the Dartmouth summer research project and explained that it is not only about correct input-output behaviour as it is mostly dominant in modern AI approaches, such as machine learning. For some the path that leads from input to output is important. Examples can be found in the fields of HCI and HRI, and generally in scenarios in which the system's behaviour needs to be explainable, anticipatory or generally speaking, understandable to humans. Human here means not the researcher or developer that has created the system, but the user, so the behaviour can be intuitively judged without the need to study AI at a university level.

Next, we have provided our definition of *human-like* behaviour, which extends the correct input-output behaviour as proposed to the Turing Test with five additional elements. We also provided a brief literature review with examples for each additional element. To the best of our

knowledge, however, no approach in the field of computer vision exists that aims at solving all of these characteristics in their entirety. From this, it became clear that *human-like* behaviours are necessary for any robotic system whose role it is to be a real assistant at home, manufacturing, service or medical setting. Further, no one has tested any of this for visuospatial behaviours at a detailed level, likely because no detailed data existed.

We continued with an explanation of which modern machine learning method applied best to our data and described our attempts at learning the same-different task. We have provided a basic introduction to reinforcement learning, setup of the environment, training strategies, simplifications of the task, and presented our results. The task was not successfully learned unless it was simplified. The foundations of reinforcement learning methodologies do not match the same-different task; humans do not learn this task; they use what they know to plan solution strategies and hypothesize and test strategies.

Chapter 6

Conclusions and Future Directions

6.1 Summary of Contributions

This thesis investigated the role of an active observer that solves a difficult yet basic visuospatial task. The intent was to discover how humans perform and to embody that performance in an artificial agent. The task was to determine if two three-dimensional real and unfamiliar objects were the same or different. We defined a particular set of objects as well as the entire experimental infrastructure for inspecting and recording human performance. Human performance yield many surprises as well as provided documentation of the reality of such problem-solving. Then, we attempted to apply the leading methods for embodiment of these human data and found that they were quite insufficient. This led to a number of proposals for future work.

In Chapter 2.2 we have presented a novel set of objects, inspired by blocks-world objects, such as the Shepard and Metzler objects. An omnipresent characteristic for most real three-dimensional objects is self-occlusion, which, as we have shown, cause difficulties for modern deep neural networks – no network was able to learn the objects. Only under rare scenarios (for instance, Hard objects from L_2) the classification accuracy was above chance. To quantify this, we introduced a metric to measure and assess the effect of self-occlusion. The need for difficult objects, such as those with self-occlusion, is crucial to push the visual system to activate visual behaviours (Section 3.4: No trial took fewer than six fixations). With this novel set of objects, we moved the traditional two-dimensional same-different task to a true three-dimensional counterpart.

We propose a three-dimensional version of the same-different task for active observation in Chapter 2. This task and our particular set of stimuli permit us to probe and examine the space of active human observation during visual problem-solving, and thus discover the depth, breadth and nature of human abilities when faced with challenging three-dimensional visuospatial tasks. Despite having seen many different instantiations also in different fields, the same-different task did not exist in three-dimensions, including active observation by humans. We provide detailed instructions for reproducibility ranging from ecological validity of the experiment, how to explain the task in a standardized form to the subjects, the control design, the stimulus, an in-depth clarification about the role of stimulus rotation, the importance of choosing different starting positions, and more. Furthermore, all custom implemented software is made publicly available.

With *PESAO* we contributed a psychophysical experimental setup for active observation – the

first of its kind. Capable of tracking precise head motion and gaze, it allows for the investigation of active visual observation in a three-dimensional world. Besides precise tracking, a strength of this system is its lightweight hardware setup that leaves the subject untethered allowing for mostly natural, free motion. In fact, only a small compute unit needs to be worn in addition to the tracking glasses. We have built *PESAO* mostly upon open-source software to also enable it to broad utility.

In Chapter 3 we used the new object set and the *PESAO* facility to conduct a large-scale user study to investigate human visual behaviours. Until now, no human experimental data was available to inform our work. We discovered complex human strategies to solve the three-dimensional same-different task. While strategies are complex, people are very good at this task, even for difficult cases – the accuracy ranges from 80% to 100%. A great deal of data acquisition occurs during all trials with at least six fixations and up to 800. We performed a complexity analysis of the task and showed that human performance is around $\mathcal{O}(n)$. Based on rich recorded data, we have identified 50 patterns of actions – Cognitive Program methods – that were repeated in various combinations. Most strikingly, no statistical change has been observed in accuracy, but one for the number of fixations, response time and head movement with increasing trials for individual subjects. This implies that subjects did not get better at the task but were more efficient.

Further, this particular task seems an excellent testbed for testing systems that purport intelligent behaviour. The Turing Test, just as an example, does not test active observation in the ways our task requires. There is no claim that this should replace it, of course; nevertheless, our data does point to a dimension of intelligence - the ability to decide how, why, when, what and where to sense the environment to best complete a task - that has not been well studied.

We have proposed that Cognitive Programs provide a flexible, dynamic composition of potential solutions. We intend to also experiment with three-dimensional "spatial relations", three-dimensional "visual search".

Lastly, we revisited the original goal of AI and made a case of why human-like behaviours are preferable and extended the correct input-output behaviour as proposed with the Turing Test with five additional criteria. We argue that these criteria are necessary for any robotic system whose role it is to be a real assistant at home, manufacturing, service or medical setting. From the modern machine learning algorithms available today, we have shown that reinforcement learning suits the three-dimensional same-different task best. However, the task was not successfully learned unless

it was simplified. Based on our findings, we formulized suggestions to inform the development of new methods and additions/modifications to existing methods.

In summary, the main contributions are:

- A novel object set for psychophysical and computational experiments, for each an example is provided in this document
- Development of the *Three-Dimensional Same-Different Task for Active Observation*
- A novel psychophysical experimental set up for active observers called *PESAO*
- Details of human visuospatial behaviours
- Representation methods for behaviours
- Showed that the existing machine learning methods are not capable of learning the *Three-Dimensional Same-Different Task for Active Observation*
- Suggestions to inform the development of new and/or modifications to existing methods

6.2 Future Direction

This dissertation aimed to answer a number of questions. However, a number of questions were also raised and need to be investigated in the future. Some of which may lead to new avenues for future research directions. Possible promising research directions are reviewed here.

6.2.1 Extensions to *PESAO*

With *PESAO* we have introduced a capable system; however, due to its design which builds primarily upon open source software, extensions are possible and might be even desired to make the system suited for different tasks beyond visuospatial ones. EEG, light sensors, full-body tracking suits, just to name a few, can be easily integrated to broaden the understanding of psychophysical experiments.

Furthermore, *PESAO* does have dependencies on proprietary software, especially driver software of the motion tracking and eye-tracking systems. It would be desirable to replace these with open source software as well, which can also be used to make the system cross-platform.

6.2.2 Examining Additional Visuospatial Tasks

With the creation of *PESAO*, we are the first to collect human visuospatial data solving a task in detail. As a testbed, we have used the well-studied same-different task, which is an instantiation of the human cognitive ability of *speeded rotation* [Carroll, 1993]. Other cognitive abilities in the realm of visual perception exist, such as *Spatial Visualization*, *Perceptual Speed*, and *Visual Search*, and with them many more tasks for each of them. In this document, we have only scratched the surface in understanding active human visual behaviours.

Additionally, what has not been studied in this work, is the effect of shading and shadowing. Both are crucial to the human visual system, which is highly dependent on light, and with it, shading and shadowing are inevitable.

The overall goal for all of this is not to solve each ability and task separately but rather to discover the common elements of a generic visual problem-solving strategy. The reality of active human visual behaviours will likely reveal many surprises to come, and with that, many challenges for how artificial agents may be developed with the same abilities.

6.2.3 Combination and Selection of Problem-Solving Strategies

We discovered that humans exhibit a variety of problem-solving strategies whose breadth and complexity are surprising and not easily handled by current methodologies. The importance of active observation is striking. These results highlight the new dimensions of visuospatial problem-solving that active observers employ.

Exactly how the problem-solving strategies are combined and selected is beyond the scope of this work but opens an exciting avenue of research. In fact, an avenue which will be investigated in follow-up work.

6.2.4 *Progressive Learning* and Human-Like Visual Behaviours

Modern machine learning, specifically reinforcement learning, was not successful in learning the three-dimensional same-different task. An avenue of research we have pointed out is *Progressive Learning*. While it remains to be tested, inspired by our findings in Chapter 3 and Chapter 5, in order to model the capabilities of human-like visual behaviours, a progression of learning sub-tasks

is necessary. The tasks are divided fine and finer, which are then used by a planning system to solve the new task. The sub-tasks correspond to small observable units of behaviour which makes them comparable to observed human behaviour, hence making them human-like.

This is different from curriculum learning for reinforcement learning [Bengio et al., 2009, Narvekar et al., 2020]. Here, the idea is to speed up the training of reinforcement learning agents by training them through a series of progressively more challenging source tasks [Narvekar and Stone, 2020]. This is in a way the opposite of *Progressive Learning*, hence different to how humans solved the three-dimensional same-different task.

With *Progressive Learning* we propose the idea to define actions and behaviours that can be assembled into cognitive programs to solve different tasks, such as the same-different task. The assembly of these programs is the interesting part that is done by the system. However, in curriculum learning, the sub-division is mostly engineered by humans and is specific to the problem at hand. In contrast, with *Progressive Learning*, the idea is to use a bottom-up approach to build higher-level programs for any tasks that share the same actions and behaviours.

Bibliography

[Abid, 2018] Abid, M. O. (2018). Cognitive Programs Memory - A Framework for Integrating Executive Control in STAR. Master's thesis, York University.

[Ackrill, 1975] Ackrill, J. L. e. a. (1975). *Categories and De interpretatione*. Clarendon Press.

[Adiv, 1989] Adiv, G. (1989). Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489.

[Aldahoul et al., 2018] Aldahoul, N., Md Sabri, A. Q., and Mansoor, A. M. (2018). Real-Time Human Detection for Aerial Captured Video Sequences via Deep Models. *Computational Intelligence and Neuroscience*, 2018(Article ID 1639561):1–14.

[Alleysson et al., 2005] Alleysson, D., Süsstrunk, S., and Hérault, J. (2005). Linear demosaicing inspired by the human visual system. *IEEE Transactions on Image Processing*, 14(4):439–449.

[Aloimonos et al., 1988] Aloimonos, J., Weiss, I., and Bandyopadhyay, A. (1988). Active vision. *International Journal of Computer Vision*, 1(4):333–356.

[Aloimonos, 1992] Aloimonos, Y. (1992). Purposive, qualitative, active vision. *CVGIP: Image Understanding*, 56(1):1–2.

[Aloimonos, 1993] Aloimonos, Y. (1993). *Introduction: Active Vision Revisited*. Psychology Press.

[Aloimonos, 1994] Aloimonos, Y. (1994). What I Have Learned. *CVGIP: Image Understanding*, 60(1):74–85.

[Aloimonos, 1995] Aloimonos, Y. (1995). Guest editorial: Qualitative vision. *International Journal of Computer Vision*, 14(2):115–117.

[Aloimonos and Shulman, 1989] Aloimonos, Y. and Shulman, D. (1989). *Integration of Visual Modules: an Extension of the Marr Paradigm*. Book, Academic Press Professional, Inc.

[Alt et al., 1988] Alt, H., Mehlhorn, K., Wagener, H., and Welzl, E. (1988). Congruence, similarity, and symmetries of geometric objects. *Discrete & Computational Geometry*, 3(3):237–256.

[Andreopoulos et al., 2011] Andreopoulos, A., Hasler, S., Wersing, H., Janssen, H., Tsotsos, J. K., and Körner, E. (2011). Active 3D object localization using a humanoid robot. *IEEE Transactions on Robotics*, 27(1):47–64.

[Andreopoulos and Tsotsos, 2012] Andreopoulos, A. and Tsotsos, J. K. (2012). On sensor bias in experimental methods for comparing interest-point, saliency, and recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):110–126.

[Andreopoulos and Tsotsos, 2013] Andreopoulos, A. and Tsotsos, J. K. (2013). 50 Years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891.

[Arsenio, 2003] Arsenio, A. (2003). Embodied vision - perceiving objects from actions. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pages 365–371.

[Arulkumaran et al., 2017] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38.

[Atkinson, 1987] Atkinson, M. D. (1987). An optimal algorithm for geometrical congruence. *Journal of Algorithms*, 8(2):159–172.

[Aubret et al., 2022] Aubret, A., Teulière, C., and Triesch, J. (2022). Embodied vision for learning object representations. *arXiv preprint arXiv:2205.06198*.

[Ayres et al., 2002] Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 429–435, New York, NY, USA. Association for Computing Machinery.

[Badia et al., 2020] Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. (2020). Agent57: Outperforming the Atari human benchmark. In Daumé, H. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 507–517.

[Bajcsy, 1985] Bajcsy, R. (1985). Active Perception vs. Passive Perception. In *Proceedings of 3rd IEEE Workshop on Computer Vision: Representation and Control*, pages 55–62, Bellaire,MI. Washington DC: IEEE Computer Society Press.

[Bajcsy, 1988] Bajcsy, R. (1988). Active Perception. *Proceedings of the IEEE*, 76(8):966–1005.

[Bajcsy, 1996] Bajcsy, R. (1996). From active perception to active cooperation - Fundamental processes of intelligent behavior. *Advances in Psychology*, 116(C):309–321.

[Bajcsy et al., 2017] Bajcsy, R., Aloimonos, Y., and Tsotsos, J. K. (2017). Revisiting active perception. *Autonomous Robots*, 42(2):177–196.

[Bajcsy and Campos, 1992] Bajcsy, R. and Campos, M. (1992). Active and exploratory perception. *CVGIP: Image Understanding*, 56(1):31–40.

[Bakhtari and Benhabib, 2007] Bakhtari, A. and Benhabib, B. (2007). An active vision system for multitarget surveillance in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1):190–198.

[Bakhtari et al., 2006] Bakhtari, A., Naish, M. D., Eskandari, M., Croft, E. A., and Benhabib, B. (2006). Active-vision-based multisensor surveillance-an implementation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(5):668–680.

[Ballard, 1989] Ballard, D. H. (1989). Behavioural constraints on animate vision. *Image and Vision Computing*, 7(1):3–9.

[Ballard, 1991] Ballard, D. H. (1991). Animate vision. *Artificial Intelligence*, 48(1):57–86.

[Ballard and Brown, 1992] Ballard, D. H. and Brown, C. M. (1992). Principles of animate vision. *CVGIP: Image Understanding*, 56(1):3–21.

[Bamber, 1969] Bamber, D. (1969). Reaction times and error rates for "same"-"different" judgments of multidimensional stimull. *Perception & Psychophysics*, 6(3):169–174.

[Bamber, 1972] Bamber, D. (1972). Reaction times and error rates for judging nominal identity of letter strings. *Perception & Psychophysics*, 12(4):321–326.

[Bamber and Paine, 1973] Bamber, D. and Paine, S. (1973). Information retrieval processes in "same"-"different" judgments of letter strings. *Attention and Performance IV*, pages 477–495.

[Barnes, 1995] Barnes, J. (1995). *The Cambridge Companion to Aristotle*. Cambridge University Press.

[Barry et al., 2014] Barry, T. J., Griffith, J. W., De Rossi, S., and Hermans, D. (2014). Meet the fribbles: Novel stimuli for use within behavioural research. *Frontiers in Psychology*, 5(FEB):1–8.

[Basile et al., 2015] Basile, B. M., Moylan, E. J., Charles, D. P., and Murray, E. A. (2015). Two-item same/different discrimination in rhesus monkeys (Macaca mulatta). *Animal Cognition*, 18(6):1221–1230.

[Belke and Meyer, 2002] Belke, E. and Meyer, A. S. (2002). Tracking the time course of multidimensional stimulus discrimination: Analyses of viewing patterns and processing times during "same"-"different" decisions. *European Journal of Cognitive Psychology*, 14(2):237–266.

[Bellemare et al., 2013] Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.

[Bengio et al., 2009] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, page 41–48.

[Bindra et al., 1968] Bindra, D., Donderi, D. C., and Nishisato, S. (1968). Decision latencies of "same" and "different" judgments. *Perception & Psychophysics*, 3(2):121–136.

[Bindra et al., 1965] Bindra, D., Williams, J. A., and Wise, J. S. (1965). Judgments of sameness and difference: Experiments on decision time. *Science*, 150(3703):1625–1627.

[Blostein and Huang, 1987] Blostein, S. and Huang, T. (1987). Quantization errors in stereo triangulation. In *Unknown Host Publication Title*, pages 325–334. IEEE.

[Boden, 2013] Boden, M. A. (2013). Mind as machine: A history of cognitive science. *Choice Reviews Online*, 44(11):44–6202–44–6202.

[Bonde et al., 2014] Bonde, U., Badrinarayanan, V., and Cipolla, R. (2014). Robust instance recognition in presence of occlusion and clutter. In *Proceedings of the European Conference on Computer Vision*, pages 520–535. Springer International Publishing.

[Borel, 1913] Borel, É. (1913). La mécanique statique et l'irréversibilité. *Journal of Physics: Theories and Applications*, 3(1):189–196.

[Brachmann et al., 2014] Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., and Rother, C. (2014). Learning 6d object pose estimation using 3d object coordinates. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Proceedings of the European Conference on Computer Vision*, pages 536–551. Springer International Publishing.

[Braß and Knauer, 2002] Braß, P. and Knauer, C. (2002). Computing the symmetries of nonconvex polyhedral objects in 3-space. *Proceedings European Workshop on Computational Geometry*, 2002:1–3.

[Brass and Knauer, 2004] Brass, P. and Knauer, C. (2004). Testing congruence and symmetry for general 3-dimensional objects. *Computational Geometry: Theory and Applications*, 27(1):3–11.

[Brockman et al., 2016] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *arXiv preprint arXiv:1606.01540*, pages 1–4.

[Browatzki et al., 2012] Browatzki, B., Tikhanoff, V., Metta, G., Bülthoff, H. H., and Wallraven, C. (2012). Active object recognition on a humanoid robot. In *2012 IEEE International Conference on Robotics and Automation*, pages 2021–2028.

[Brown and Austin, 2021] Brown, M. F. and Austin, B. P. (2021). Bees and abstract concepts. *Current Opinion in Behavioral Sciences*, 37:140–145.

[Burgundand and Marsolek, 2000] Burgundand, D. and Marsolek, C. J. (2000). Viewpoint-invariant and viewpoint-dependent object recognition in dissociable neural subsystems. *North-Holland Mathematics Studies*, 187(C):141–170.

[Caglioti, 2001] Caglioti, V. (2001). An entropic criterion for minimum uncertainty sensing in recognition and localization. I. Theoretical and conceptual aspects. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(2):187–196.

[Carroll, 1993] Carroll, J. B. (1993). *Human cognitive abilities: A survey of factor-analytic studies*. Cambridge University Press.

[Chen et al., 2011] Chen, S., Li, Y., and Kwok, N. M. (2011). Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377.

[Chen et al., 2008] Chen, S., Li, Y. F., Wang, W., and Zhang, J. (2008). *Active sensor planning for multiview vision tasks*, volume 1. Springer.

[Chen and Li, 2004] Chen, S. Y. and Li, Y. F. (2004). Automatic Sensor Placement for Model-Based Robot Vision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):393–408.

[Chen and Li, 2005] Chen, S. Y. and Li, Y. F. (2005). Vision sensor planning for 3-D model acquisition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):894–904.

[Chu and Chung, 2002] Chu, G. W. and Chung, M. J. (2002). Autonomous selection and modification of camera configurations using visibility and manipulability measures. *Journal of Robotic Systems*, 19(5):219–230.

[Clowes, 1971] Clowes, M. B. (1971). On seeing things. *Artificial intelligence*, 2(1):79–116.

[Community, 2018] Community, B. O. (2018). Blender - a 3d modelling and rendering package. `http://www.blender.org`.

[Coumans and Bai, 2016] Coumans, E. and Bai, Y. (2016). PyBullet, a Python module for physics simulation for games, robotics and machine learning. `https://pybullet.org`.

[Crowley et al., 1992] Crowley, J., Krotkov, E., and Brown, C. (1992). Active computer vision: A tutorial. In *IEEE International Conference on Robotics and Automation*.

[Damen et al., 2018] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. (2018). Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision*.

[Das and Ahuja, 1996] Das, S. and Ahuja, N. (1996). Active surface estimation: integrating coarse-to-fine image acquisition and estimation from multiple cues. *Artificial Intelligence*, 83(2):241–266.

[Davies, 2016] Davies, N. (2016). Can robots handle your healthcare? *Engineering and Technology*, 11(9):58–61.

[Davis and Goldwater, 2021] Davis, T. and Goldwater, M. (2021). Using model-based neuroimaging to adjudicate structured and continuous representational accounts in same-different categorization and beyond. *Current Opinion in Behavioral Sciences*, 37:103–108.

[de Melo et al., 2021] de Melo, C. M., Torralba, A., Guibas, L., DiCarlo, J., Chellappa, R., and Hodgins, J. (2021). Next-generation deep learning based on simulators and synthetic data. *Trends in Cognitive Sciences*, 26(2):174–187.

[Deinzer et al., 2009] Deinzer, F., Derichs, C., Niemann, H., and Denzler, J. (2009). A framework for actively selecting viewpoints in object recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):765–799.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[Dickinson et al., 1997] Dickinson, S. J., Christensen, H. I., Tsotsos, J. K., and Olofsson, G. (1997). Active object recognition integrating attention and viewpoint control. *Computer Vision and Image Understanding*, 67(3):239–260.

[Dickinson et al., 2009] Dickinson, S. J., Leonardis, A., Schiele, B., and Tarr, M. J. (2009). *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press.

[Dickinson et al., 1999] Dickinson, S. J., Wilkes, D. R., and Tsotsos, J. K. (1999). A Computational Model of View Degeneracy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):673–689.

[Dickmanns, 1997] Dickmanns, E. D. (1997). Vehicles Capable of Dynamic Vision. In *International Joint Conference on Artificial Intelligence*, pages 1577–1592.

[Dickmanns, 2007] Dickmanns, E. D. (2007). *Dynamic vision for perception and control of motion*. Springer Science and Business Media.

[Dickmanns and Graefe, 1988] Dickmanns, E. D. and Graefe, V. (1988). Dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):223–240.

[Dollár et al., 2012] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761.

[Donnon et al., 2005] Donnon, T., DesCôteaux, J. G., and Violato, C. (2005). Impact of cognitive imaging and sex differences on the development of laparoscopic suturing skills. *Canadian Journal of Surgery*, 48(5):387–393.

[Dosovitskiy et al., 2015] Dosovitskiy et al., A. (2015). FlowNet: Learning Optical Flow with Convolutional Networks Alexey. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766.

[DuTell et al., 2021] DuTell, V., Gibaldi, A., Focarelli, G., Olshausen, B., and Banks, M. S. (2021). Integrating high fidelity eye, head and world tracking in a wearable device. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*. Association for Computing Machinery.

[Dyer, 1973] Dyer, F. N. (1973). Same and different judgments for word-color pairs with "irrelevant" words or colors: evidence for word-code comparisons. *Journal of Experimental Psychology*, 98(1):102.

[Egeth, 1966] Egeth, H. E. (1966). Parallel versus serial processes in multidimensional stimulus discrimination. *Perception & Psychophysics*, 1(4):245–252.

[Ellenrieder et al., 2005] Ellenrieder, M. M., Krüger, L., Stößel, D., and Hanheide, M. (2005). A versatile model-based visibility measure for geometric primitives. In *Scandinavian Conference on Image Analysis*, pages 669–678. Springer.

[Eltoft and DeFigueiredo, 1995] Eltoft, T. and DeFigueiredo, R. J. P. (1995). Illumination control as a means of enhancing image features in active vision systems. *Proceedings of the IEEE Transactions on Image Processing*, 4(11):1520–1530.

[Espeholt et al., 2018] Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., and Others (2018). Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceedings of the International Conference on Machine Learning*, pages 1407–1416.

[Ess et al., 2009] Ess, A., Schindler, K., Leibe, B., and Van Gool, L. (2009). Improved multi-person tracking with active occlusion handling. In *ICRA Workshop on People Detection and Tracking*, volume 2. Citeseer.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338.

[Farell, 1977] Farell, B. (1977). Encoding and comparisons in "same"-"different" judgments. *Unpublished doctoral thesis). McGill University, Montréal, Canada.*

[Farell, 1985] Farell, B. (1985). " same"–" different" judgments: A review of current controversies in perceptual comparisons. *Psychological Bulletin*, 98(3):419.

[Fei-Fei et al., 2006] Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611.

[Fellbaum, 2010] Fellbaum, C. (2010). *WordNet*, pages 231–243. Springer Netherlands, Dordrecht.

[Findlay et al., 2003] Findlay, J. M., Findlay, J. M., Gilchrist, I. D., and Others (2003). *Active vision: The Psychology of Looking and Seeing*. Number 37. Oxford University Press.

[Fink, 2004] Fink, M. (2004). Object classification from a single example utilizing class relevance metrics. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

[Fiore et al., 2008] Fiore, L., Somasundaram, G., Drenner, A., and Papanikolopoulos, N. (2008). Optimal camera placement with adaptation to dynamic scenes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 956–961.

[Flandin and Chaumette, 2001] Flandin, G. and Chaumette, F. (2001). Vision-based control using probabilistic geometry for objects reconstruction. In *Proceedings of the IEEE Conference on Decision and Control*, volume 5, pages 4152–4157.

[Flandin and Chaumette, 2002] Flandin, G. and Chaumette, F. (2002). Visual data fusion for objects localization by active vision. In *Proceedings of the European Conference on Computer Vision*, pages 312–326. Springer.

[Fleuret et al., 2011] Fleuret, F., Li, T., Dubout, C., Wampler, E. K., Yantis, S., and Geman, D. (2011). Comparing machines and humans on a visual categorization test. *Proceedings of the National Academy of Sciences of the United States of America*, 108(43):17621–17625.

[Fodor and Pylyshyn, 1988] Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

[Fournier-Viger et al., 2016] Fournier-Viger, P., Lin, J. C.-W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., and Lam, H. T. (2016). The SPMF open-source data mining library version 2. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–40. Springer.

[Funke et al., 2021] Funke, C. M., Borowski, J., Stosio, K., Brendel, W., Wallis, T. S., and Bethge, M. (2021). Five points to check when comparing visual perception in humans and machines. *Journal of Vision*, 21(3):1–23.

[Gauthier and Tarr, 1997] Gauthier, I. and Tarr, M. J. (1997). Becoming a 'Greeble' expert: Exploring mechanisms for face recognition. *Vision Research*, 37(12):1673–1682.

[Gay-Bellile et al., 2010] Gay-Bellile, V., Bartoli, A., and Sayd, P. (2010). Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):87–104.

[Geirhos et al., 2017] Geirhos, R., Janssen, D. H. J., Schütt, H. H., Rauber, J., Bethge, M., and Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*.

[Giannotti et al., 2007] Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. (2007). Trajectory pattern mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 330–339. Association for Computing Machinery.

[Gibson, 1979] Gibson, J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifling, Boston.

[Girshick et al., 2011] Girshick, R., Felzenszwalb, P., and McAllester, D. (2011). Object detection with grammar models. In *Advances in Neural Information Processing Systems*, volume 24, pages 442–450. Curran Associates, Inc.

[Goldstein, 1981] Goldstein, E. B. (1981). The Ecology of J. J. Gibson's Perception. *Leonardo*, 14(3):191.

[Gouaillier et al., 2009] Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2009). Mechatronic design of NAO humanoid. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 769–774.

[Grauman et al., 2022] Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S. K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E. Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Gebreselasie, A., González, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolář, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Ruiz, P., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X.,

Yagi, T., Zhao, Z., Zhu, Y., Arbeláez, P., Crandall, D., Damen, D., Farinella, G. M., Fuegen, C., Ghanem, B., Ithapu, V. K., Jawahar, C. V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H. S., Rehg, J. M., Sato, Y., Shi, J., Shou, M. Z., Torralba, A., Torresani, L., Yan, M., and Malik, J. (2022). Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 18995–19012.

[Graves, 1993] Graves, R. (1993). *The Greek Myths: The Complete Edition*. Penguin UK.

[Haarnoja et al., 2018] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Proceedings of the International Conference on Machine Learning*, 5:2976–2989.

[Hadamard, 1902] Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, pages 49–52.

[Han and Charles, 2019] Han, Y. and Charles, L. (2019). *A New Method to Solve Same-different Problems with Few-shot A New Method to Solve Same-different Problems with Few-shot Learning Learning*. PhD thesis, The University of Western Ontario.

[Hanson, 1978] Hanson, A. (1978). *Computer Vision Systems*. Elsevier.

[Harding, 2018] Harding, B. (2018). *A single process model of the same-different task*. PhD thesis, Université d'Ottawa/University of Ottawa.

[Hastie et al., 2005] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). Reviews-the elements of statistical learning: data mining, inference and prediction. *Mathematical Intelligencer*, 27(2):83–84.

[Hausamann et al., 2020] Hausamann, P., Sinnott, C., and MacNeilage, P. R. (2020). Positional head-eye tracking outside the lab: An open-source solution. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*. Association for Computing Machinery.

[He et al., 2017] He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision*.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Proceedings of the European Conference on Computer Vision*, pages 630–645, Cham. Springer International Publishing.

[Hessel et al., 2018] Hessel, M., Modayil, J., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., and Silver, D. (2018). Rainbow: Combining Improvements in Deep Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[Highlander and Rodriguez, 2016] Highlander, T. and Rodriguez, A. (2016). Very efficient training of convolutional neural networks using fast fourier transform and overlap-and-add. *arXiv preprint arXiv:1601.06815*.

[Hinterstoisser et al., 2013] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., and Navab, N. (2013). Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. *Proceedings of Asian Conference of Computer Vision*, pages 548–562.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

[Ho-Phuoc, 2018] Ho-Phuoc, T. (2018). CIFAR10 to Compare Visual Recognition Performance between Deep Neural Networks and Humans. *arXiv preprint arXiv:1811.07270*.

[Hodaň et al., 2017] Hodaň, T., Haluza, P., Obdrzalek, Š., Matas, J., Lourakis, M., and Zabulis, X. (2017). T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 880–888.

[Hodge and Kamel, 2003] Hodge, L. and Kamel, M. (2003). An agent-based approach to multisensor coordination. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 33(5):648–661.

[Horn et al., 1986] Horn, B., Klaus, B., and Horn, P. (1986). *Robot Vision*. MIT Press.

[Horn and Weldon, 1987] Horn, B. K. P. and Weldon, J. (1987). Computationally efficient methods for recovering translational motion. *Proceedings of the IEEE International Conference on Computer Vision*, 871000010(X):2–11.

[Hsiao et al., 2010] Hsiao, E., Collet, A., and Hebert, M. (2010). Making specific features less discriminative to improve point-based 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2653–2660.

[Hsiao and Hebert, 2014] Hsiao, E. and Hebert, M. (2014). Occlusion reasoning for object detectionunder arbitrary viewpoint. *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1803–1815.

[Huang and Mutlu, 2016] Huang, C. M. and Mutlu, B. (2016). Anticipatory robot control for efficient human-robot collaboration. *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, April:83–90.

[Huang and Blonstein, 1985] Huang, T. and Blonstein, M. (1985). Robust algorithms for computing three dimensional motion from image sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Huffman, 1971] Huffman, D. A. (1971). Impossible object as nonsense sentences. *Machine Intelligence*, 6:295–324.

[Hummel, 1987] Hummel, R. (1987). Solving ill-conditioned problems by minimizing equation error. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 527–533.

[Ilg et al., 2018] Ilg, E., Saikia, T., Keuper, M., and Brox, T. (2018). Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *Proceedings of the European Conference on Computer Vision*.

[Itti et al., 1998] Itti, L., Koch, C., and Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

[Jalal et al., 2019] Jalal, M., Spjut, J., Boudaoud, B., and Betke, M. (2019). Sidod: A synthetic image dataset for 3d object pose recognition with distractors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Jhuang, 2007] Jhuang, H. (2007). *A Biologically Inspired system for action recognition*. PhD thesis, Massachusetts Institute of Technology.

[Jiang et al., 1996] Jiang, X., Yu, K., and Bunke, H. (1996). Detection of rotational and involutional symmetries and congruity of polyhedra. *The Visual Computer*, 12(4):193–201.

[Jiang and Bunke, 1991] Jiang, X.-Y. and Bunke, H. (1991). Determination of the symmetries of polyhedra and an application to object recognition. In *Workshop on Computational Geometry*, pages 113–121. Springer.

[Johnson, 1970] Johnson, N. F. (1970). The role of chunking and organization in the process of recall. In *Psychology of Learning and Motivation*, volume 4, pages 171–247. Elsevier.

[Johnson et al., 2017] Johnson et al., J. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910. IEEE.

[Jonnalagadda et al., 2003] Jonnalagadda, K., Lumia, R., Starr, G., and Wood, J. (2003). Viewpoint selection for object reconstruction using only local geometric features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 2116–2122.

[Jordan and Mitchell, 2015] Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

[Juliani et al., 2018] Juliani, A., Berges, V.-P., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., and Others (2018). Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*.

[Kabra et al., 2019] Kabra, R., Burgess, C., Matthey, L., Kaufman, R. L., Greff, K., Reynolds, M., and Lerchner, A. (2019). Multi-Object Datasets. `https://github.com/deepmind/multi_object_datasets`.

[Kaelbling, Leslie Pack and Littman, Michael L and Moore, 1996] Kaelbling, Leslie Pack and Littman, Michael L and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

[Kaneko et al., 2019] Kaneko, K., Kaminaga, H., Sakaguchi, T., Kajita, S., Morisawa, M., Kumagai, I., and Kanehiro, F. (2019). Humanoid robot HRP-5P: An electrically actuated humanoid robot with high-power and wide-range joints. *Proceedings of the IEEE Robotics and Automation Letters*, 4(2):1431–1438.

[Katz and Wright, 2021] Katz, J. S. and Wright, A. A. (2021). Issues in the comparative cognition of same/different abstract-concept learning. *Current Opinion in Behavioral Sciences*, 37:29–34.

[Kelion, 2019] Kelion, L. (2019). DeepMind AI achieves Grandmaster status at Starcraft 2. https://www.bbc.com/news/technology-50212841.

[Kempka et al., 2016] Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Jaśkowski, W. (2016). Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*, pages 1–8.

[Khamis et al., 2017] Khamis, M., Hoesl, A., Klimczak, A., Reiss, M., Alt, F., and Bulling, A. (2017). EyeScout: Active eye tracking for position and movement independent gaze interaction with large public displays. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 155–166.

[Kim et al., 2018] Kim, J., Ricci, M., and Serre, T. (2018). Not-So-CLEVR: learning same–different relations strains feedforward neural networks. *Interface Focus*, 8: 20180011:1–13.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*, pages 1–15.

[Kirousis and Papadimitriou, 1988] Kirousis, L. M. and Papadimitriou, C. H. (1988). The complexity of recognizing polyhedral scenes. *Journal of Computer and System Sciences*, 37(1):14–38.

[Kjellin et al., 2010] Kjellin, A., Pettersson, L. W., Seipel, S., and Lind, M. (2010). Evaluating 2D and 3D visualizations of spatiotemporal information. *ACM Transactions on Applied Perception*, 7(3):1–23.

[Kober et al., 2014] Kober, J., Bagnell, J. A., and Peters, J. (2014). Reinforcement Learning in Robotics: A Survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

[Koch, 2015] Koch, G. (2015). Siamese Neural Networks for One-Shot Image Recognition. Master's thesis, University of Toronto.

[Kokhlikyan et al., 2020a] Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. (2020a). Captum: A unified and generic model interpretability library for PyTorch. *arXiv preprint arXiv:2009.07896*.

[Kokhlikyan et al., 2020b] Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. (2020b). VEDB headset V1. `http://visualexperiencedatabase.org/research.html`.

[Koporec and Pers, 2019] Koporec, G. and Pers, J. (2019). Deep learning performance in the presence of significant occlusions - an intelligent household refrigerator case. In *Proceedings of the IEEE International Conference on Computer Vision Workshop*.

[Korbach et al., 2021] Korbach, C., Solbach, M. D., Memmesheimer, R., Paulus, D., and Tsotsos, J. K. (2021). Next-Best-View Estimation based on Deep Reinforcement Learning for Active Object Classification. *arXiv preprint arXiv:2110.06766*.

[Kortylewski et al., 2021] Kortylewski, A., Liu, Q., Wang, A., Sun, Y., and Yuille, A. (2021). Compositional Convolutional Neural Networks: A Robust and Interpretable Model for Object Recognition Under Occlusion. *International Journal of Computer Vision*, 129(3):736–760.

[Kothari et al., 2020] Kothari, R., Yang, Z., Kanan, C., Bailey, R., Pelz, J. B., and Diaz, G. J. (2020). Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities. *Scientific Reports*, 10(1):1–18.

[Kothe, 2014] Kothe, C. (2014). Lab streaming layer (LSL). `https://github.com/sccn/labstreaminglayer`.

[Kotseruba and Tsotsos, 2017] Kotseruba, I. and Tsotsos, J. K. (2017). STAR-RT: Visual attention for real-time video game playing. *arXiv preprint arXiv:1711.09464*.

[Kotseruba and Tsotsos, 2020] Kotseruba, I. and Tsotsos, J. K. (2020). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1):17–94.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[Krueger, 1973] Krueger, L. (1973). Effect of stimulus frequency on speed of same-different judgments. *Attention and Performance IV*, pages 497–506.

[Kunic, 2017] Kunic, T. (2017). Cognitive program compiler. Master's thesis, York University.

[Kusuda, 2010] Kusuda, Y. (2010). The use of robots in the Japanese food industry. *Industrial Robot*, 37(6):503–508.

[Kutulakos and Dyer, 1994] Kutulakos, K. N. and Dyer, C. R. (1994). Recovering shape by purposive viewpoint adjustment. *International Journal of Computer Vision*, 12(2):113–136.

[Kuznetsova et al., 2020] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., and Ferrari, V. (2020). The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *International Journal of Computer Vision*, 128(7):1956–1981.

[Kwok et al., 2006] Kwok, N. M., Liu, D. K., and Dissanayake, G. (2006). Evolutionary computing based mobile robot localization. *Engineering Applications of Artificial Intelligence*, 19(8):857–868.

[Lai et al., 2014] Lai, K., Bo, L., and Fox, D. (2014). Unsupervised feature learning for 3D scene labeling. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3050–3057.

[Lake et al., 2011] Lake, B. M., Salakhutdinov, R., Gross, J., and Tenenbaum, J. B. (2011). One shot learning of simple visual concepts. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 33(33).

[Lampert et al., 2008] Lampert, C. H., Blaschko, M. B., and Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[Lampert et al., 2009] Lampert, C. H., Nickisch, H., and Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958.

[Lanctot et al., 2019] Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S., Hennes, D., Morrill, D., Muller, P., Ewalds, T., Faulkner, R., Kramár, J., De Vylder, B., Saeta, B., Bradbury, J., Ding, D., Borgeaud, S., Lai, M., Schrittwieser, J., Anthony, T., Hughes, E., Danihelka, I., and Ryan-Davis, J. (2019). OpenSpiel: A Framework for Reinforcement Learning in Games. *arXiv preprint arXiv:1908.09453*, pages 1–27.

[Landy et al., 2012] Landy, M. S., Maloney, L. T., and Pavel, M. (2012). *Exploratory Vision: The Active Eye*. Springer Science and Business Media.

[Lapan, 2018] Lapan, M. (2018). *Deep Reinforcement Learning Hands-On: Apply Modern RL Methods, with Deep Q-networks, Value Iteration, Policy Gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd.

[Larochelle et al., 2008] Larochelle, H., Erhan, D., and Bengio, Y. (2008). Zero-data learning of new tasks. *Proceedings of the National Conference on Artificial Intelligence*, 2:646–651.

[Lázaro-Gredilla et al., 2019] Lázaro-Gredilla, M., Lin, D., Guntupalli, J. S., and George, D. (2019). Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs. *Science Robotics*, 4(26).

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[LeCun et al., 2010] LeCun, Y., Kavukcuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. *Proceedings of the IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, pages 253–256.

[Levine et al., 2018] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *International Journal of Robotics Research*, 37(4-5):421–436.

[Li et al., 2019] Li, C., Zia, M. Z., Tran, Q. H., Yu, X., Hager, G. D., and Chandraker, M. (2019). Deep Supervision with Intermediate Concepts. *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1828–1843.

[Li et al., 2017] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816.

[Li et al., 2009] Li, L.-J., Socher, R., and Fei-Fei, L. (2009). Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2036–2043.

[Lim et al., 2007] Lim, S.-N., Davis, L., and Mittal, A. (2007). Task scheduling in large camera networks. In *Asian Conference on Computer Vision*, pages 397–407. Springer.

[Lin et al., 2014] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Proceedings of the European Conference on Computer Vision*, 8693 LNCS(PART 5):740–755.

[Link and Tindall, 1971] Link, S. W. and Tindall, A. D. (1971). Speed and accuracy in comparative judgments of line length. *Perception & Psychophysics*, 9(3):284–288.

[Livingstone and Spacek, 1996] Livingstone, D. and Spacek, L. (1996). *A Behavioural Vision System for Search and Motion Tracking*. na.

[Loewe and Shaughnessy, 1999] Loewe, M. and Shaughnessy, E. L. (1999). *The Cambridge history of ancient China: From the origins of civilization to 221 BC*. Cambridge University Press.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the IEEE International Conference on Computer Vision*, 2(8):1150–1157.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Luursema et al., 2012] Luursema, J. M., Verwey, W. B., and Burie, R. (2012). Visuospatial ability factors and performance variables in laparoscopic simulator training. *Learning and Individual Differences*, 22(5):632–638.

[Lynn and Olsen, 2018] Lynn, G. and Olsen, C. (2018). Interview with Greg Lynn: Forward-Thinking Land Drones. *Technology Architecture and Design*, 2(2):143–145.

[Manderson et al., 2020] Manderson, T., Gamboa, J. C., Wapnick, S., Shkurti, F., Meger, D., and Dudek, G. (2020). Vision-Based Goal-Conditioned Policies for Underwater Navigation in the Presence of Obstacles. In *Proceedings of Robotics: Science and Systems*.

[Marcus, 2003] Marcus, G. F. (2003). *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

[Marr, 1982] Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

[Marr and Nishihara, 1978] Marr, D. and Nishihara, H. K. (1978). Visual Information Processing: Artificial Intelligence and the Sensorium of Sight. *MIT Technology Review*, 81(1):2–23.

[Martinho and Kacelnik, 2016] Martinho, A. and Kacelnik, A. (2016). Ducklings imprint on the relational concept of "same or different". *Science*, 353(6296):286–288.

[Maspero, 1895] Maspero, G. (1895). *Manual of Egyptian Archaeology and Guide to the Study of Antiquities in Egypt: For the Use of Students and Travellers*. H. Grevel and Company.

[Matthey et al., 2017] Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. (2017). dSprites: Disentanglement testing Sprites dataset. `https://github.com/deepmind/dsprites-dataset/`.

[Mayer et al., 2016] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[McCarthy et al., 2006] McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (2006). A proposal for the Dartmouth summer research project on artificial intelligence. *AI Magazine*, 27(4):12–14.

[McKevitt, 1997] McKevitt, P. (1997). Ai: The tumultuous history of the search for artificial intelligence. *The British Journal for the History of Science*, 30(1):101–121.

[Meger et al., 2011] Meger, D., Wojek, C., Little, J. J., and Schiele, B. (2011). Explicit Occlusion Reasoning for 3D Object Detection. In *Proceedings of the British Machine Vision Conference*, pages 1–11. Citeseer.

[Megreya and Burton, 2006] Megreya, A. M. and Burton, A. M. (2006). Unfamiliar faces are not faces: Evidence from a matching task. *Memory & Cognition*, 34(4):865–876.

[Mertsching and Schmalz, 1999] Mertsching, B. and Schmalz, S. (1999). Active Vision Systems. In *Handbook of Computer Vision and Applications: Systems and Applications (Vol. 3)*, pages 197–2019.

[Meudec, 2021] Meudec, R. (2021). tf-explain. `https://github.com/sicara/tf-explain`.

[Miao et al., 2019] Miao, J., Wu, Y., Liu, P., Ding, Y., and Yang, Y. (2019). Pose-guided feature alignment for occluded person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*.

[Michel et al., 2017] Michel, F., Kirillov, A., Brachmann, E., Krull, A., Gumhold, S., Savchynskyy, B., and Rother, C. (2017). Global hypothesis generation for 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*, volume 1. McGraw-hill New York.

[Moor, 2006] Moor, J. (2006). Artificial Intelligence Conference : The Next Fifty Years. *AI Magazine*, 27(4):87–91.

[Mostofi, 2011] Mostofi, Y. (2011). Compressive cooperative sensing and mapping in mobile networks. *IEEE Transactions on Mobile Computing*, 10(12):1769–1784.

[Motai and Kosaka, 2008] Motai, Y. and Kosaka, A. (2008). Hand–eye calibration applied to viewpoint selection for robotic vision. *IEEE Transactions on Industrial Electronics*, 55(10):3731–3741.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT press.

[Naish et al., 2003] Naish, M. D., Croft, E. A., and Benhabib, B. (2003). Coordinated dispatching of proximity sensors for the surveillance of manoeuvring targets. *Robotics and Computer-Integrated Manufacturing*, 19(3):283–299.

[Narvekar et al., 2020] Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. (2020). Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21:1–50.

[Narvekar and Stone, 2020] Narvekar, S. and Stone, P. (2020). Generalizing Curricula for Reinforcement Learning. *Proceedings of the International Conference on Machine Learning*.

[Nickerson, 1965] Nickerson, R. S. (1965). Response times for "same"-"different" judgments. *Perceptual and Motor Skills*, 20(1):15–18.

[Nickerson, 1967] Nickerson, R. S. (1967). Categorization time with categories defined by disjunctions and conjunctions of stimulus attributes. *Journal of Experimental Psychology*, 73(2):211.

[Nickerson, 1969] Nickerson, R. S. (1969). 'Same'-'different'response times: A model and a preliminary test. *Acta Psychologica*, 30:257–275.

[Nickerson and Pew, 1973] Nickerson, R. S. and Pew, R. W. (1973). Visual pattern matching: An investigation of some effects of decision task, auditory codability, and spatial correspondence. *Journal of Experimental Psychology*, 98(1):36.

[Nikishin et al., 2018] Nikishin, E., Izmailov, P., Athiwaratkun, B., Podoprikhin, D., Garipov, T., Shvechikov, P., Vetrov, D., and Wilson, A. G. (2018). Improving Stability in Deep Reinforcement Learning with Weight Averaging. *UAI workshop on Uncertainty in Deep Learning*, pages 1–5.

[Nilsson and Fikes, 1971] Nilsson, N. J. and Fikes, R. E. (1971). STRIPS : A New Approach to the Application of Theorem Proving to. *Artificial Intelligence*, 8(October):189–208.

[Norman, 2005] Norman, D. A. (2005). Robots in the home: what might they do? *Interactions*, 12(2):65.

[Ognibene and Demiris, 2013] Ognibene, D. and Demiris, Y. (2013). Towards active event recognition. *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2495–2501.

[Ouyang and Wang, 2012] Ouyang, W. and Wang, X. (2012). A discriminative deep model for pedestrian detection with occlusion handling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3258–3265.

[Palmer, 1999] Palmer, S. E. (1999). *Vision Science: Photons to Phenomenology*. MIT press.

[Parodi et al., 1998] Parodi, P., Lancewicki, R., Vijh, A., and Tsotsos, J. K. (1998). Empirically-derived estimates of the complexity of labeling line drawings of polyhedral scenes. *Artificial Intelligence*, 105:47–75.

[Pei et al., 2004] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440.

[Pepikj et al., 2013] Pepikj, B., Stark, M., Gehler, P., and Schiele, B. (2013). Occlusion patterns for object class detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Petrov, 2009] Petrov, A. A. (2009). Symmetry-based methodology for decision-rule identification in same-different experiments. *Psychonomic Bulletin & Review*, 16(6):1011–1025.

[Petrusic et al., 1978] Petrusic, W. M., Varro, L., and Jamieson, D. G. (1978). Mental rotation validation of two spatial ability tests. *Psychological Research*, 40(2):139–148.

[Radwan et al., 2013] Radwan, I., Dhall, A., and Goecke, R. (2013). Monocular image 3d human pose estimation under self-occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*.

[Rasouli and Tsotsos, 2014] Rasouli, A. and Tsotsos, J. K. (2014). Visual saliency improves autonomous visual search. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 111–118.

[Rasouli and Tsotsos, 2015] Rasouli, A. and Tsotsos, J. K. (2015). Attention and Sensor Planning in Autonomous Robotic Visual Search. Master of applied science, York University.

[Ravichandiran, 2020] Ravichandiran, S. (2020). *Deep Reinforcement Learning with Python: Master Classic RL, Deep RL, Distributional RL, Inverse RL, and more with OpenAI Gym and TensorFlow*. Packt Publishing.

[Reddy et al., 2019] Reddy, N. D., Vo, M., and Narasimhan, S. G. (2019). Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Ren et al., 2022] Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. (2022). A Survey of Deep Active Learning. *ACM Computing Surveys*, 54(9).

[Ricci et al., 2018] Ricci, M., Kim, J., and Serre, T. (2018). Same-different problems strain convolutional neural networks. *arXiv preprint arXiv:1802.03390*.

[Rivlin et al., 1992] Rivlin, E., Aloimonos, Y., and Rosenfeld, A. (1992). Purposive recognition: an active and qualitative approach. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 225–240. International Society for Optics and Photonics.

[Roberts, 1960] Roberts, L. G. (1960). Pattern recognition with an adaptive network. *Proceedings of the Institute of Radio Engineers*, 48(3).

[Roberts, 1963] Roberts, L. G. (1963). *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology.

[Robla-Gomez et al., 2017] Robla-Gomez, S., Becerra, V. M., Llata, J. R., Gonzalez-Sarabia, E., Torre-Ferrero, C., and Perez-Oria, J. (2017). Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments. *IEEE Access*, 5:26754–26773.

[Rosheim, 2006] Rosheim, M. (2006). *Leonardo's Lost Robots*. Springer Science & Business Media.

[Roy et al., 2000] Roy, S. D., Chaudhury, S., and Banerjee, S. (2000). Isolated 3D object recognition through next view planning. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(1):67–76.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *Proceedings of the International Journal of Computer Vision*, 115(3):211–252.

[Sahin et al., 2019] Sahin, C., Garcia-Hernando, G., Sock, J., and Kim, T. K. (2019). Instance- and Category-Level 6D Object Pose Estimation. In *Advances in Computer Vision and Pattern Recognition*, pages 243–265.

[Salvemini et al., 2011] Salvemini, E., Fumarola, F., Malerba, D., and Han, J. (2011). FAST sequence mining based on sparse id-lists. *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, pages 316–325.

[Sandler et al., 2018] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

[Schaie, 1989] Schaie, K. W. (1989). Perceptual speed in adulthood: Cross-sectional and longitudinal studies. *Psychology and Aging*, 4(4):443–453.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, pages 1–12.

[Settles, 2009] Settles, B. (2009). Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences*.

[Shah et al., 2018] Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer.

[Shankar et al., 2021] Shankar, B., Sinnott, C., Binaee, K., Lescroart, M. D., and Macneilage, P. (2021). Ergonomic Design Development of the Visual Experience Database Headset. *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, pages 1–4.

[Shepard and Metzler, 1971] Shepard, R. N. and Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703.

[Sheridan, 2016] Sheridan, T. B. (2016). Human-Robot Interaction. *Human Factors*, 58(4):525–532.

[Shin et al., 2015] Shin, E., Lee, H., Yoo, S. A., and Chong, S. C. (2015). Training improves the capacity of visual working memory when it is adaptive, individualized, and targeted. *PLoS ONE*, 10(4):1–14.

[Shorten and Khoshgoftaar, 2019] Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1).

[Shu et al., 2012] Shu, G., Dehghan, A., Oreifej, O., Hand, E., and Shah, M. (2012). Part-based multiple-person tracking with partial occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1815–1821.

[Shubina and Tsotsos, 2010] Shubina, K. and Tsotsos, J. K. (2010). Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535–547. Special issue on Intelligent Vision Systems.

[Shwartz-Ziv and Tishby, 2017] Shwartz-Ziv, R. and Tishby, N. (2017). Opening the Black Box of Deep Neural Networks via Information. *arXiv preprint arXiv:1703.00810*, pages 1–19.

[Sigurdardottir et al., 2018] Sigurdardottir, H. M., Fridriksdottir, L. E., Gudjonsdottir, S., and Kristjánsson, Á. (2018). Specific problems in visual cognition of dyslexic readers: Face discrimination deficits predict dyslexia over and above discrimination of scrambled faces and novel objects. *Cognition*, 175(March):157–168.

[Sigurdsson et al., 2018] Sigurdsson, G. A., Gupta, A., Schmid, C., Farhadi, A., and Alahari, K. (2018). Charades-Ego: A Large-Scale Dataset of Paired Third and First Person Videos. *arXiv preprint arXiv:1804.09626*, pages 1–3.

[Silver et al., 2021] Silver, D., Singh, S., Precup, D., and Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, 299:103535.

[Silverman and Goldberg, 1975] Silverman, W. P. and Goldberg, S. L. (1975). Further confirmation of same vs. different processing differences. *Perception & Psychophysics*, 17(2):189–193.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Siu and Murphy, 2018] Siu, C. R. and Murphy, K. M. (2018). The development of human visual cortex and clinical implications. *Eye and Brain*, 10:25–36.

[Slaney and Thiébaux, 2001] Slaney, J. and Thiébaux, S. (2001). Blocks World revisited. *Artificial Intelligence*, 125(1-2):119–153.

[Snodgrass, 1972] Snodgrass, J. G. (1972). Matching patterns vs matching digits: The effect of memory dependence and complexity on "same"-"different" reaction times. *Perception & Psychophysics*, 11(5):341–349.

[Solbach et al., 2018] Solbach, M. D., Voland, S., Edmonds, J., and Tsotsos, J. K. (2018). Random polyhedral scenes: An image generator for active vision system experiments. *arXiv preprint arXiv:1803.10100*.

[Srinath et al., 2021] Srinath, R., Emonds, A., Wang, Q., Lempel, A. A., Dunn-Weiss, E., Connor, C. E., and Nielsen, K. J. (2021). Early Emergence of Solid Shape Coding in Natural and Deep Network Vision. *Current Biology*, 31(1):51–65.

[Srivastava et al., 2015] Srivastava, S., Zilberstein, S., Gupta, A., Abbeel, P., and Russell, S. (2015). Tractability of planning with loops. *Proceedings of the National Conference on Artificial Intelligence*, 5:3393–3401.

[Stabinger et al., 2016] Stabinger, S., Rodríguez-Sánchez, A., and Piater, J. (2016). 25 years of CNNS: Can we compare to human abstraction capabilities? *Proceedings of the International Conference on Artificial Neural Networks*, 9887 LNCS:380–387.

[Stanley, 1988] Stanley, R. P. (1988). Differential posets. *Journal of the American Mathematical Society*, 1(4):919–961.

[Stone et al., 2021] Stone, S. A., Boser, Q. A., Dawson, T. R., Vette, A. H., Hebert, J. S., Pilarski, P. M., and Chapman, C. S. (2021). Sub-centimeter 3d gaze vector accuracy on real-world tasks: An investigation of eye and motion capture calibration routines. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*. Association for Computing Machinery.

[Sugihara, 1984] Sugihara, K. (1984). An n log n algorithm for determining the congruity of polyhedra. *Journal of Computer and System Sciences*, 29(1):36–47.

[Suppa and Hirzinger, 2007] Suppa, M. and Hirzinger, G. (2007). Multisensorielle Exploration von Roboterarbeitsräumen (Multisensory Exploration of Robot Workspaces). *TM-Technisches Messen*, 74(3):139–146.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction Second Edition*. MIT press.

[Sutton et al., 1998] Sutton, R. S., Barto, A. G., and Others (1998). *Introduction to Reinforcement Learning*. MIT press Cambridge.

[Swain and Stricker, 1993] Swain, M. J. and Stricker, M. A. (1993). Promising directions in active vision. *International Journal of Computer Vision*, 11(2):109–126.

[Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

[Taigman et al., 2014] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verificatio. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

[Tan and Le, 2019] Tan, M. and Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the International Conference on Machine Learning*, volume 97, pages 6105–6114.

[Tarabanis et al., 1994] Tarabanis, K., Tsai, R., and Allen, P. (1994). Analytical characterization of the feature detectability constraints of resolution, focus, and field-of-view for vision sensor planning. *CVGIP: Image Understanding*, 59(3):340–358.

[Tarabanis et al., 1996] Tarabanis, K., Tsai, R. Y., and Kaul, A. (1996). Computing occlusion-free viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):279–292.

[Tarabanis et al., 1995] Tarabanis, K. A., Allen, P. K., and Tsai, R. Y. (1995). A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1):86–104.

[Taylor et al., 2021] Taylor, A. T., Berrueta, T. A., and Murphey, T. D. (2021). Active learning in robotics: A review of control principles. *Mechatronics*, 77(May):102576.

[Taylor, 1976] Taylor, D. A. (1976). Effect of identity in the multiletter matching task. *Journal of Experimental Psychology: Human Perception and Performance*, 2(3):417.

[Terzopoulos and Rabie, 1995] Terzopoulos, D. and Rabie, T. (1995). Animat vision: Active vision in artificial animals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 801–808.

[Tippett et al., 1965] Tippett, J., Borkowitz, D. A., Clapp, L. C., Koester, C. J., and Vanderburgh Jr, A. (1965). Optical and electro-optical information processing. Technical report, Massachusetts Institute of Technology Cambridge.

[Tkacz-Domb and Yeshurun, 2018] Tkacz-Domb, S. and Yeshurun, Y. (2018). The size of the attentional window when measured by the pupillary response to light. *Scientific Reports*, 8(1):1–7.

[Todorov et al., 2012] Todorov, E., Erez, T., and Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. *IEEE International Conference on Intelligent Robots and Systems*, pages 5026–5033.

[Trefethen and Bau III, 1997] Trefethen, L. N. and Bau III, D. (1997). *Numerical Linear Algebra*, volume 50. Siam.

[Triebel and Burgard, 2008] Triebel, R. and Burgard, W. (2008). Recovering the shape of objects in 3d point clouds with partial occlusions. In *Field and Service Robotics*, pages 13–22. Springer.

[Trucco et al., 1997] Trucco, E., Umasuthan, M., Wallace, A. M., and Roberto, V. (1997). Model-based planning of optimal sensor placements for inspection. *IEEE Transactions on Robotics and Automation*, 13(2):182–194.

[Tsotsos, 1987] Tsotsos, J. K. (1987). A Complexity Level Analysis of Vision. *International Journal of Computer Vision*, 1(4):303–320.

[Tsotsos, 1992] Tsotsos, J. K. (1992). On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2):127–141.

[Tsotsos, 2010] Tsotsos, J. K. (2010). Re-Visiting Visual Routines: A White Paper. Technical report, York University, Technical Report CSE-2010-11, October 1.

[Tsotsos et al., 2021] Tsotsos, J. K., Abid, O., Kotseruba, I., and Solbach, M. D. (2021). On the control of attentional processes in vision. *Cortex*, 137:305–329.

[Tsotsos and Kruijne, 2014] Tsotsos, J. K. and Kruijne, W. (2014). Cognitive programs: software for attention's executive. *Frontiers in Psychology*, 5.

[Tsotsos, 1989] Tsotsos, J. Y. U. (1989). The Complexity of Perceptual Search Tasks. *Proceedings of the International Joint Conference on Artificial Intelligence*, 89:1571–1577.

[Turing, 1950a] Turing, A. M. (1950a). Computing machinery and intelligence. *Mind*, 59(236):433–460.

[Turing, 1950b] Turing, A. M. (1950b). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.

[Tversky, 1969] Tversky, B. (1969). Pictorial and verbal encoding in a short-term memory task. *Perception & Psychophysics*, 6(4):225–233.

[Ullman, 1987] Ullman, S. (1987). Visual routines. In Fischler, M. A. and Firschein, O., editors, *Readings in Computer Vision*, pages 298–328. Morgan Kaufmann, San Francisco (CA).

[Vallat, 2018] Vallat, R. (2018). Pingouin: statistics in Python. *The Journal of Open Source Software*, 3(31):1026.

[Van Opstal, 2021] Van Opstal, F. (2021). The same-different task as a tool to study unconscious processing. *Current Opinion in Behavioral Sciences*, 37:35–40.

[Van Opstal and Verguts, 2011] Van Opstal, F. and Verguts, T. (2011). The origins of the numerical distance effect: the same–different task. *Journal of Cognitive Psychology*, 23(1):112–120.

[Vázquez, 2007] Vázquez, P.-P. (2007). Automatic light source placement for maximum visual information recovery. In *Computer Graphics Forum*, volume 26, pages 143–156. Wiley Online Library.

[Vedaldi and Zisserman, 2009] Vedaldi, A. and Zisserman, A. (2009). Structured output regression for detection with partial truncation. In *Advances in Neural Information Processing Systems*, volume 22, pages 1–9. Curran Associates, Inc.

[Vieville, 2012] Vieville, T. (2012). *A few steps towards 3d active vision*. Springer Science and Business Media, 33 edition.

[Vinyals et al., 2019] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

[Viola and Jones, 2004] Viola, P. and Jones, M. J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, pages 137–154.

[Voulodimos et al., 2018] Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018:13.

[Wang et al., 2013] Wang, T., He, X., and Barnes, N. (2013). Learning structured hough voting for joint object detection and occlusion reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Wang et al., 2009] Wang, X., Han, T. X., and Yan, S. (2009). An hog-lbp human detector with partial occlusion handling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 32–39.

[Wanzel et al., 2003] Wanzel, K. R., Hamstra, S. J., Caminiti, M. F., Anastakis, D. J., Grober, E. D., and Reznick, R. K. (2003). Visual-spatial ability correlates with efficiency of hand motion and successful surgical performance. *Surgery*, 134(5):750–757.

[Well et al., 1975] Well, A. D., Pollatsek, A., and Schindler, R. M. (1975). Facilitation of both "same" and "different" judgments of letter strings by familiarity of letter sequence. *Perception & Psychophysics*, 17(5):511–520.

[Wheeler and Ikeuchi, 1995] Wheeler, M. D. and Ikeuchi, K. (1995). Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):252–265.

[Wilkes, 1994] Wilkes, D. R. (1994). *Active object recognition*. PhD thesis, University of Toronto.

[Wilkes and Tsotsos, 1993] Wilkes, D. R. and Tsotsos, J. K. (1993). Behaviors for Active Object Recognition. *Intelligent Robots and Computer Vision*, 2055(XII):225–239.

[Wloka et al., 2016] Wloka, C., Yoo, S.-A., Sengupta, R., Kunic, T., and Tsotsos, J. (2016). Psychophysical Evaluation of Saliency Algorithms. *Journal of Vision*, 16(12):1291.

[Woodcroft and Others, 1851] Woodcroft, B. and Others (1851). *The Pneumatics of Hero of Alexandria: From the Original Greek*. Charles Whittingham.

[Wu and Nevatia, 2009] Wu, B. and Nevatia, R. (2009). Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *International Journal of Computer Vision*, 82(2):185–204.

[Xing et al., 2009] Xing, J., Ai, H., and Lao, S. (2009). Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1200–1207.

[Yamane et al., 2008] Yamane, Y., Carlson, E. T., Bowman, K. C., Wang, Z., and Connor, C. E. (2008). A neural code for three-dimensional object shape in macaque inferotemporal cortex. *Nature Neuroscience*, 11(11):1352–1360.

[Yang et al., 2019] Yang, J., Ren, Z., Xu, M., Chen, X., Crandall, D., Parikh, D., and Batra, D. (2019). Embodied amodal recognition: Learning to move to perceive objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2040–2050.

[Ye and Tsotsos, 1999] Ye, Y. and Tsotsos, J. K. (1999). Sensor Planning for 3D Object Search. *Computer Vision and Image Understanding*, 73(2):145–168.

[Zaki, 2001] Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60.

[Zheng et al., 2015] Zheng, W.-S., Li, X., Xiang, T., Liao, S., Lai, J., and Gong, S. (2015). Partial person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*.

[Zhu et al., 2021] Zhu, F., Zhu, Y., Lee, V., Liang, X., and Chang, X. (2021). Deep learning for embodied vision navigation: A survey. *arXiv preprint arXiv:2108.04097*.

Appendix A

Consent Form

Attached is a copy of the consent form signed by the subjects participating in the research conducted in Chapter 3.

Informed Consent Form

Date: January 6th, 2020
Study Name: Visuospatial Functionality for Active Observers: The Same-Different Task
Researcher: Markus D. Solbach (solbach@eecs.yorku.ca), Lassonde Building (Room 3054), York University, 4700 Keele St, Toronto ON. M3J 1P3 Canada
Purpose of the Research: The same-different task is relevant for any kind of robot whose role is to be a real assistant in the home or in manufacturing or medical setting. It also has clear scientific value with the goal of understanding how humans accomplish visual attention tasks. Specifically, we hope that the same-different task illustrates the ways in which attention and cognition connect. These then point to new avenues of research that might illuminate the overall cognitive architecture of spatial cognition.
What You Will Be Asked to Do in the Research: As the same-different task is designed, the participant has to wear eye-tracking glasses and a set of passive markers mounted on the glasses. The glasses are connected to a small mobile processing unit that can be clipped on a belt. After the subject is equipped with the hardware, the glasses have to get calibrated, which requires the subject to look at a calibration marker for a few seconds. The experiment is run in a controlled, clean environment in which two objects are presented in the center of it. The subject is asked to determine whether the two objects are the same or different. Same means that they have the exact same appearance (shape, size and color). The subject is asked to perform the task as precise as possible, where timing is secondary. A total of 18 different object pairings are presented. Between each trial the subject is asked to approach one of the three starting positions (explained before the start of the experiment), face the curtains and wait for the start signal of the next trial. While waiting for the start signal the subject has the opportunity to relax and rest their eyes. Important to note is that the subject equipped with the hardware is completely untethered with the environment to avoid any tripping hazards. The subject is allowed to move around freely. The estimated time commitment will be approximately 40 minutes.
Risks and Discomforts: We do not foresee any risks or discomfort from your participation in the research. However, some participants might encounter minor discomfort wearing the hardware equipment. In any case, the subject is permitted to take a break whenever needed or to quit the experiment at any time.
Benefits of the Research: It is almost universal to regard attention as the facility that permits an agent, human or machine, to give priority processing resources to relevant stimuli while ignoring the irrelevant. The reality of how this might manifest itself throughout all the forms of perceptual and cognitive processes possessed by humans, however, is not as clear. Here we examine this reality with a broad perspective in order to highlight the myriad ways that attentional processes impact both perception and cognition. The same-different task exhibits sufficient complexity to illustrate the ways in which attention and cognition connect. These then point to new avenues of research that might illuminate the overall cognitive architecture of spatial cognition.
Voluntary Participation and Withdrawal: Your participation in the study is completely voluntary, and you may choose to stop participating at any time. Your decision not to volunteer, to stop participating, or to refuse to answer particular questions will not influence the nature of the ongoing relationship you may have with the researchers or study staff and nature of your relationship with York University either now or in the future. In the event you withdraw from the study, all associated data collected will be immediately destroyed wherever possible.
Confidentiality: All information you supply during the research will be held in confidence. Confidentiality will be provided to the fullest extent possible by law.
The data collected in this research project may be used – in an anonymized form - by members of the research team in subsequent research investigations exploring similar lines of inquiry. Such projects will still undergo ethics review by the HPRC, our institutional REB. Any secondary use of anonymized data by the research team will be treated with the same degree of confidentiality and anonymity as in the original research project.
We intend to use the data publicly for demonstration purposes (presentations, webpage, etc.). The data will be provided identifiable. However, personal information such as gender, age, and so on, are treated anonymously. Your name will not be recorded, and your data will be stored using an anonymous ID number.
Questions About the Research? If you have questions about the research in general or about your role in the study, please feel free to contact Markus D. Solbach by e-mail (solbach@eecs.yorku.ca). This research has received ethics review and approval by the Human Participants Review Sub-Committee, York University's Ethics Review Board, and conforms to the standards of the Canadian Tri-Council Research Ethics guidelines. If you have any questions about this process or about your rights as a participant in the study, please contact the Sr. Manager & Policy Advisor for the Office of Research Ethics, 5th Floor, Kaneff Tower, York University (telephone 416-736-5914 or e-mail ore@yorku.ca).

Legal Rights and Signatures:

I,_____, consent to participate in *Visuospatial Functionality for Active Observers: The Same-Different Task* conducted by *Markus D. Solbach*. I have understood the nature of this project and wish to participate. I am not waiving any of my legal rights by signing this form. My signature below indicates my consent.

Signature_____ **Date**_____
Participant

Signature_____ **Date**_____
Principal Investigator

Additional consent

You must seek additional consent by including check boxes or requesting additional signatures for the following:

1. **Video recording or use of photographs**

I, _____, consent to the use of images of me (including photographs, video and other moving images), my environment and property in the following ways (please check all that apply):

In academic articles	☐ N	☐ Y
In print, digital and slide form	☐ N	☐ Y
In academic presentations	☐ N	☐ Y
In media	☐ N	☐ Y
In thesis materials	☐ N	☐ Y
In public dataset	☐ N	☐ Y

Signature_____ **Date**_____
Participant Name: