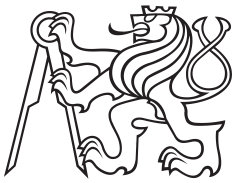
Master's Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Sophia the Robot - Development of a Software Extension - Detection and Reaction to Being Photographed by a Phone Camera

Master's Thesis

Bc. Jakub Šura

Supervisor: David Hanson, Ph.D.
Supervisor at CTU: Ing. Petr Pošík, Ph.D.
Field of study: Cybernetics and Robotics
Subfield: Cybernetics and Robotics
May 2024

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Šura Jakub** Personal ID number: **495796**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Sophia the Robot - Development of a Software Extension - Detection and Reaction to Being Photographed by a Phone Camera

Master’s thesis title in Czech:

Sophia the Robot - vývoj rozšíření softwaru - detekce a reakce na focení telefonní kamerou

Guidelines:

"Sophia the Robot" developed by Hanson Robotics is a realistic humanoid robot capable of displaying humanlike expressions and interacting with people. Her main methods of perception are an RGB-D camera and an array of microphones. She is designed for research, education, and entertainment, and helps promote public discussion about AI ethics and the future of robotics. When people interact with Sophia, they often want to take pictures with/of her. The goal is to write an addition to her code that would allow her to recognize the intent of a person to take a picture, recognize the phone being used to take the picture, and react accordingly, i.e., smile, look at the camera, wave, say "cheese", etc. This feature would make her seem more aware and lifelike, one of the goals of her ongoing development. The code will be written mainly in Python and incorporated as a new ROS node. The inputs will be the video stream from her RGB-D camera and the audio from her microphones. The proposed methods include object and pose detection and tracking in video, speech recognition, decision making, and will be explored upon further. The output will be a ROS message commanding her to look at the position of the detected camera, its coordinates, and a reaction to take (smile, speak, wave). The success will be rated by objective metrics (e.g., response time, gaze accuracy) and subjective metrics (e.g., how good does the picture look).

Bibliography / sources:

[1] Hanson, David, et al. “Hanson Robotics.” GitHub. Accessed November 3, 2023. https://github.com/hansonrobotics/.
[2] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics vol. 7, May 2022.
[3] Cao, Zhe, et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields.” arXiv, 30 May 2019. arXiv.org, https://doi.org/10.48550/arXiv.1812.08008.
[4] Wang, Chien-Yao, et al. “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors.” arXiv, 6 July 2022. arXiv.org, https://doi.org/10.48550/arXiv.2207.02696.

Name and workplace of master’s thesis supervisor:

David Hanson, Ph.D. CEO of Hanson Robotics Limited, Hong Kong SAR China

Name and workplace of second master’s thesis supervisor or consultant:

Ing. Petr Pošík, Ph.D. Department of Cybernetics FEE

Date of master’s thesis assignment: **03.11.2023** Deadline for master's thesis submission: **24.05.2024**

Assignment valid until: **22.09.2024**

David Hanson, Ph.D. Supervisor’s signature	prof. Dr. Ing. Jan Kybic Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Acknowledgements

This thesis would not have been possible without the significant support and opportunity provided by David Hanson, Ph.D., and the team at Hanson Robotics Limited. Their help and expertise were crucial to my research, especially Vytas Krisciunas who provided invaluable guidance and mentorship throughout the project. I am equally thankful to Kath Yeung who made this collaboration possible and provided continuous encouragement.

Lastly, my heartfelt thanks go to my family, whose endless support and encouragement have been my constant source of strength and motivation.

Declaration

I hereby declare that I have written this thesis by myself and that I have listed all information sources in accordance with Methodical Guideline No. 2/2024 on Compliance with Ethical Principles *(CTU in Prague, 19. 02. 2024)*.

As per the updated guidelines, I declare that tools generally recognized as *"Artificial Intelligence"* (specifically various large language models developed by *OpenAI*) have been used in the development and writing of this work. Their primary uses were drafting, proofreading, and formatting both written word and code.

Prague, 25. May 2024

Abstract

This thesis presents an enhancement of Sophia the Robot, a humanoid platform renowned for its human-like expressions and interactions. The project focuses on developing an advanced human-robot interaction system aimed at enhancing multi-party awareness, with a particular emphasis on adding the ability to detect being photographed and to respond in a natural manner.

Essential to this initiative is a comprehensive software overhaul, supported by emerging machine learning technologies, such as zero-shot detection and multimodal large language models, and a novel gaze control behavior module that mimics human-like dynamic interactions. The implementation utilizes the Robot Operating System (ROS), enhancing adaptability for use in other robotic systems. Concurrently, proposed hardware modifications support the enhanced sensory requirements.

The system was tested and evaluated in real-world scenarios, with user evaluations designed to compare the responses to typical human reactions. This research advances humanoid robotics by demonstrating a customizable framework for improving robotic responsiveness in human-centric applications.

Keywords: Human-robot interaction, Humanoid robot, Computer vision, Scene analysis, Attention and gaze control system, Multimodal interaction, Robot Operating System (ROS)

Abstrakt

Tato práce se věnuje rozšíření možností Sophie, humanoidního robota známého realistickým expresivním obličejem a interakcemi. Projekt se zaměřuje na vývoj pokročilého systému vnímání a pozornosti, který má za cíl zlepšit skupinovou interakci, přičemž klade důraz na schopnost detekovat situaci kdy je robot fotografován a reagovat přirozeným způsobem.

Zásadní součástí projektu je komplexní přepracování softwaru, aplikace moderních technologií strojového učení, jako jsou modely pro zero-shot detekci a multimodální velké jazykové modely, spolu s nově navrženým modulem pro řízení pozornosti, který imituje dynamickou humanoidní interakci. Implementace využívá Robot Operating System (ROS), což umožňuje adaptaci systému pro jiné robotické platformy. Současně jsou navrženy úpravy hardwaru, které podporují rozšířené senzorické požadavky.

Systém byl testován a hodnocen na reálných situacích, kde hodnocení uživatelů sloužilo k porovnání reakcí robota s typickými lidskými reakcemi. Tato práce přináší pokrok v oblasti humanoidní robotiky tím, že demonstruje flexibilní systém pro zlepšení reaktivity robotů v aplikacích zaměřených na interakci s člověkem.

Klíčová slova: Interakce člověka s robotem, Humanoidní robot, Strojové vidění, Analýza scény, Systém řízení pozornosti a pohledu, Multimodální interakce, Robot Operating System (ROS)

Contents

1 Introduction **1**

1.1 Background 1

1.1.1 Sophia the Robot 1

1.1.2 Humanoid Robots in Society . 2

1.1.3 Challenges in Social Robotics . 2

1.2 Motivation and Goals 3

1.2.1 Enhancing Sophia's Autonomy 3

1.2.2 Improving Interaction Capabilities through Photogenic Behavior 4

1.3 Thesis Structure 5

2 System Architecture, Development, and Implementation **7**

2.1 Hardware Layer 8

2.1.1 Starting State 8

2.1.2 Objectives 12

2.1.3 Implementation 12

2.2 Perception Layer 16

2.2.1 Starting State 16

2.2.2 Objectives 17

2.2.3 Implementation 17

2.3 Attention Layer 25

2.3.1 Starting State 25

2.3.2 Objectives 26

2.3.3 Implementation 28

2.4 Reasoning Layer 37

2.4.1 Starting State 37

2.4.2 Objectives 39

2.4.3 Implementation 39

2.5 Behavior Layer 41

2.5.1 Starting State 41

2.5.2 Objectives 42

2.5.3 Implementation 42

3 Experiments, Evaluation, and Discussion **49**

3.1 Experimental Setup 49

3.2 Questionnaire and Feedback 52

3.3 Results and Discussion 54

3.4 Comparisons 56

3.5 Challenges and Limitations 58

4 Conclusion and Future Work 59

4.1 Contributions 59

4.2 Future Development 61

A Bibliography 63

B Code Archive 69

Figures

1.1 Sophia displaying a range of emotions. From left to right: happy, angry, surprised, winking. *(Source: Author and Hanson Robotics Limited)* 2

1.2 People interacting with Sophia and taking pictures and videos of her. *(Source: Hanson Robotics Limited)* . 4

2.1 Sophia's body components in a basic configuration. *(Source: Author, with permission from Hanson Robotics Limited)* 8

2.2 Intel RealSense D435i, Sophia's main camera for face detection and depth perception mounted on her chest. *(Source: Author and Hanson Robotics Limited)* 9

2.3 A close-up of Sophia's eye cameras, used for eye contact correction. *(Source: Author and Hanson Robotics Limited)* 10

2.4 HRSDK WebUI, Sophia's frontend interface for operators to control her behavior and monitor her sensors. *(Source: Hanson Robotics Limited)* 11

2.5 iFlytek Far-Field Microphone Array Module, a suitable directional microphone array for Sophia's sound source localization system. *(Source: Hiwonder)* [1] 13

2.6 Sophia's 6 DoF movable torso prototype. *(Source: Hanson Robotics Limited)* 14

2.7 Illustration of the proposed peripheral vision system with a fisheye camera and torso yaw mechanism. *(Source: Author)* 15

2.8 Illustration of the proposed peripheral hearing system employing a directional microphone array and torso yaw mechanism. *(Source: Author)* 15

2.9 See3CAM-CU81, an example of a suitable fisheye camera for Sophia's peripheral vision system. *(Source: E-con Systems)* [2] 15

2.10 Visualization of keypoints estimated by the YOLOv8-Pose model. *(Source: [3])* 19

2.11 A depiction of all 80 classes in the COCO dataset. *(Source: [4])* 24

2.12 Example output of the Perception Layer in the HRSDK WebUI. *(Source: Author with permission from Hanson Robotics Limited)* 25

2.13 Example of a quadratic distance penalty calculation. *(Source: Author)* 32

2.14 Illustration of eFOV angle. *(Source: Author)* 32

2.15 Example of attention span fluctuations over time and the resulting attention switching. *(Source: Author)* 33

2.16 Illustration of the audio dropoff angle. *(Source: Author)* 36

2.17 Illustration of the attention inheritance mechanism. *(Source: Author)* 37

2.18 Screenshot of the attention layer configuration page in the HRSDK WebUI. This configuration is used for the `posing` behavior state. *(Source: Author with permission from Hanson Robotics Limited)* 38

2.19 State hierarchy of the HSM for Sophia's behavior layer. Added states have been higlighted. *(Source: Author)* 43

2.20 Screenshot of the HRSDK WebUI showing the configuration of the picture-taking state triggers. *(Source: Author with permission from Hanson Robotics Limited)* 45

2.21 Terminal output showing the state transition logic in action. The combination of a detected picture-taking device and an appropriate keyword triggers the `selfie` state. *(Source: Author)* . . . 46

2.22 Screenshot of the HRSDK WebUI showing behavior state customization options. The appropriate animations for posing for a photo have been chosen. *(Source: Author with permission from Hanson Robotics Limited)* 47

2.23 Sophia posing for a picture, waving.*(Source: Author and Hanson Robotics Limited)* 48

3.1 Testing the `gaze_angle_penalty` mechanism. From left to right: *Person 1* and *Person 2*. *(Source: Author and Hanson Robotics Limited)* 51

3.2 Testing the `attention_inheritance` mechanism. From left to right: *Person 1* and *Person 2*. *(Source: Author and Hanson Robotics Limited)* 53

3.3 Sophia's creator, Dr. David Hanson, testing the new Photogenic Behavior module. 55

Chapter 1

Introduction

1.1 Background

1.1.1 Sophia the Robot

Sophia the Robot, developed by Hanson Robotics Limited, is recognized for her state-of-the-art human-like expressive face 1.1 and for being one of the first robots of her kind to achieve celebrity status. Unveiled in 2016 [5], Sophia has become an icon in the realm of social robotics, distinguished by her ability to mimic human emotions and engage in meaningful dialogues. Her sophisticated appearance and advanced AI integration make her a key figure in exploring the intersection of technology and social interaction [6].

Notable historical milestones for Sophia include being granted citizenship by Saudi Arabia [7], speaking at the United Nations [8], and appearing on popular television shows which have significantly raised public interest in robotics and AI [9]. These events have not only showcased her capabilities but have also sparked discussions on the future of robots in society [10].

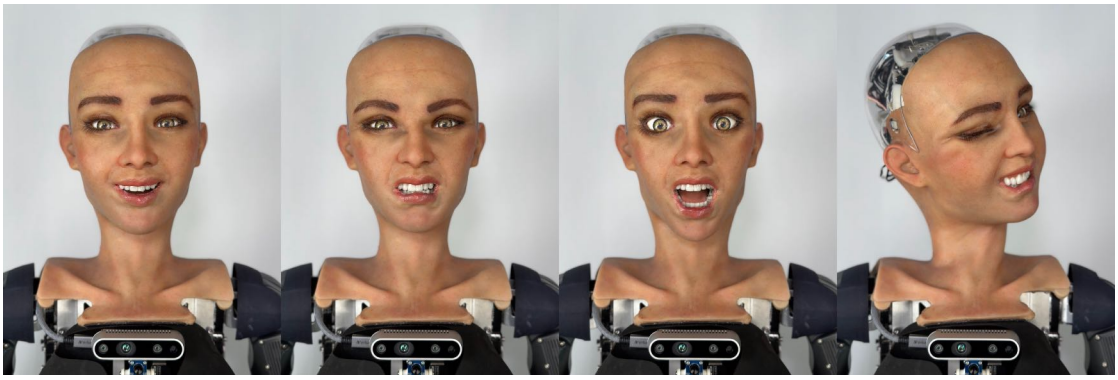


Figure 1.1: Sophia displaying a range of emotions. From left to right: happy, angry, surprised, winking. *(Source: Author and Hanson Robotics Limited)*

1.1.2 Humanoid Robots in Society

Sophia is part of a growing family of humanoid robots that are increasingly integrated into society [11]. These robots are designed to interact with humans in various settings, such as healthcare [12], education [13], entertainment [14], and customer service [15]. Mukherjee, Baral, et al., systematically reviewed humanoid robots in healthcare, highlighting their growing acceptance and diverse applications, which informs the potential pathways for Sophia's integration in similar environments [12]. Kirstein, Risager, et al., implemented social robots in educational institutions, demonstrating increased engagement and learning outcomes, which are critical considerations for Sophia's development [13].

Social robots are distinct from traditional industrial robots in that they are intended to engage with people in a human-like manner, often through verbal and non-verbal communication [16]. In addition to task-performing capabilities, social robots must be able to display socially acceptable behavior. This includes recognizing social cues, understanding context, and responding appropriately to human actions. These abilities are essential for establishing trust and rapport with users, which are crucial for successful human-robot interactions [17].

1.1.3 Challenges in Social Robotics

One of the key challenges in social robotics is the ability of robots to understand and respond to social cues - nonverbal signals that people use in communicating their intentions and attention. These cues include facial expressions, body language, and gaze direction.

For example, maintaining eye contact with someone indicates attentiveness and interest in the conversation. Social robots thus require a mechanism that is able to control attention, gaze, and behavior on the basis of social cues. This mechanism, known as the *gaze control system* (GCS), is responsible for determining where the robot should look and how it should respond to social cues [18]. The GCS is a critical component of social robots, as it enables them to engage in natural and socially appropriate interactions with humans [19].

Zaraki, Mazzei, et al., designed and evaluated a human-specific GCS, which partially aligns with the enhancements of Sophia's GCS [20], as well as a similar work by Aliasghari, Taheri, et al. [21]

1.2 Motivation and Goals

1.2.1 Enhancing Sophia's Autonomy

Sophia's public interactions are highlighted by her capabilities in maintaining eye contact, recognizing faces, and engaging in human-like conversations, making her a sought-after participant at high-profile events worldwide [22] and a popular figure in the media [23]. Despite these advanced features, her interactions are not fully autonomous. Some of her behavior during public appearances is controlled by human operators who direct her focus — whether towards a camera, an audience member, or other points of interest. We can see an example of this in Figure 1.2 where we see multiple people interacting with Sophia at the same time and taking pictures and videos of her.

Sophia's interactivity is limited by her current software, especially by her GCS, which is primarily configured to track human faces with no other stimuli being considered. Effective in controlled environments, this system does not adequately handle the dynamic and unpredictable nature of unstructured social settings. For instance, at public events, Sophia is often confronted with numerous simultaneous stimuli, such as reporters, cameras, and excited attendees, which her existing system cannot prioritize in a natural human-like manner.

The initiative to advance Sophia's autonomy aims to transform her to a more independent entity, and to move away from reliance on operators. This transformation is

Figure 1.2: People interacting with Sophia and taking pictures and videos of her. *(Source: Hanson Robotics Limited)*

essential for improving the authenticity of her interactions and enabling her to manage complex, multi-party environments on her own.

1.2.2 Improving Interaction Capabilities through Photogenic Behavior

The choice to enhance Sophia's autonomy and interaction capabilities specifically through improvements in her ability to detect and react to photograph-taking scenarios serves as both a focal point for this research and a stepping stone for broader enhancements. The choice of this feature, aptly named (*Photogenic Behavior*), is motivated by its visibility and frequent occurrence at public events, making it a tangible and relatable improvement. However, the underlying enhancements to her system architecture are designed with a wider vision.

Sophia is a multipurpose platform, thus the new system needs to be flexible, scalable, and transparent, allowing for the integration of additional features and behaviors in the future. The goal was to create a system that is not only capable of detecting and reacting

to photograph-taking scenarios but also able to adapt to a wide range of social situations, making Sophia more versatile and engaging.

This work also contributes to previous research, enabling previously only theoretical or virtually tested systems to be implemented and evaluated in real-world scenarios, such as [24], in which Shen, Mo, Krisciunas, Hanson, and Shi investigated intention estimation via gaze for robot guidance in hierarchical tasks, in virtual environment.

1.3 Thesis Structure

This thesis is organized into four main chapters that delineate the development process, the architecture enhancements, and the comprehensive testing of the modifications applied to Sophia the Robot, culminating in a discussion of the implications and future directions of this research.

- **System Architecture, Development, and Implementation**
  This chapter is the core of the thesis, detailing the systematic enhancements made to Sophia's architecture. It is divided into several sections, each focusing on different layers of the new system:
  - *Hardware Layer*: Discusses the existing hardware components and the integration of new hardware designed to expand Sophia's sensory capabilities and interaction potential.
  - *Perception Layer*: Introduces new detection models and algorithms that enable Sophia to perceive and understand her environment.
  - *Attention Layer*: Details the attention management system that directs Sophia's focus dynamically based on her interaction context.
  - *Reasoning Layer*: Explores Sophia's AI agents and their role in providing context-aware information to the GCS.
  - *Behavior Layer*: Describes the mechanisms through which Sophia's behavior is regulated and adapted based on the perceived stimuli and the desired interaction outcomes. Covers the user-facing mechanisms through which Sophia responds, providing a natural interaction experience.
- **Experiments, Evaluation, and Discussion**
  This chapter presents the methodology and results of the experiments conducted to test the newly implemented systems in real-world scenarios. It includes:

- *Experimental Setup*: Describes the conditions and parameters of the tests conducted.
- *Questionnaire and Feedback*: Summarizes the feedback received from users and operators during testing and evaluation.
- *Results and Discussion*: Analyzes the results of the experiments and evaluates the effectiveness of the enhancements made to Sophia's system.
- *Comparisons*: Compares the performance of the new system with the original one and discusses the improvements observed. Additionally, it compares this system with other similar systems in the field of social robotics.
- *Challenges and Limitations*: Acknowledges the obstacles encountered during the project and the constraints of the current system. Touches on the possible future enhancements that could address these challenges.

- **Conclusion and Future Work**
  The final chapter synthesizes the contributions of the thesis and discusses the broader impacts of the research. It summarizes the key points of the study and outlines potential future research directions that could further enhance Sophia's capabilities or apply the findings to other robotic systems.

  - *Summary of Contributions*: Concisely recaps the enhancements made to Sophia and their significance within the field of social robotics.
  - *Future Research Directions*: Proposes avenues for continued development based on the results and experiences gained during this project.

Chapter 2

System Architecture, Development, and Implementation

As concluded in the previous chapter, the best way to implement Photogenic Behavior is a major overhaul of the current human-robot interaction (HRI) system, with focus on the gaze control system (GCS) and the behavior system.

The project has been split the into five main interconnected layers: *hardware*, *perception*, *attention*, *reasoning*, and *behavior*. Modeling human behavior, this system serves to collect visual and auditory information, process them, modify the robot's behavior based on the perceived inputs and inner reasoning, and react appropriately, both verbally and physically.

For the specific purpose of Photogenic Behavior, the GCS cannot simply track the nearest and/or the centremost picture-taking device or person, it needs to understand the difference between distinct scenarios (such as selfie, regular picture taking, candid shot, or video recording), adjust the *behavior* state accordingly, which in turn tunes the *attention* model. The *attention* layer then decides which detected object or person to track based on a complex set of features extracted by the *perception* layer and returns a single point to gaze at, and customized gaze parameters.

This chapter describes these layers, the logic and research behind their design, and their implementation in practice.

2.1 Hardware Layer

2.1.1 Starting State

- **Mechanical Components**

  The mechanical components of Sophia include her face, neck, upper torso, and arms (Figure 2.1). These components are responsible for her expressive capabilities, allowing her to mimic human facial expressions and gestures.

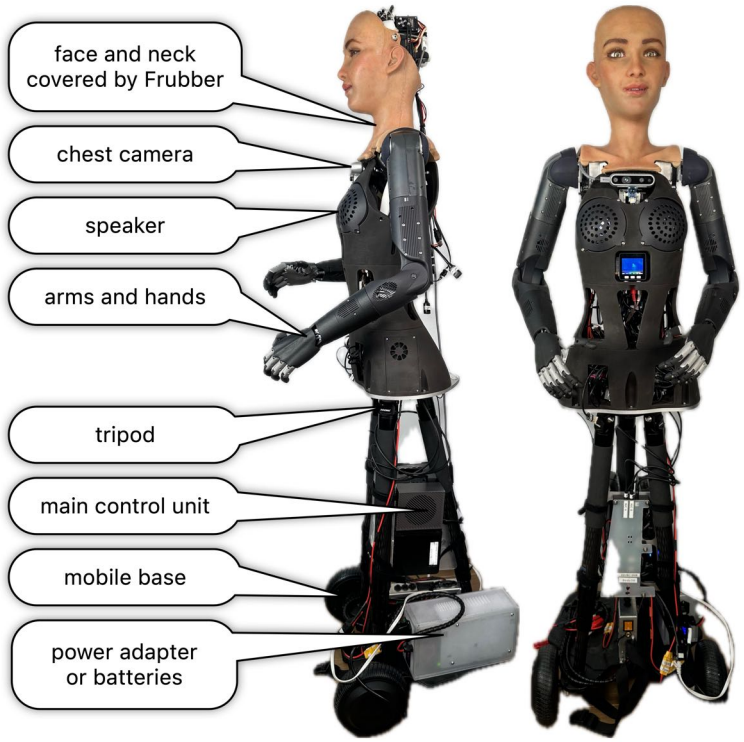


Figure 2.1: Sophia's body components in a basic configuration. *(Source: Author, with permission from Hanson Robotics Limited)*

Sophia's head is the most expressive part of her body, featuring a realistic human-like face with 32 facial actuators that enable her to produce a wide range of expressions. Her skin is made of a patented elastomer-based material called Frubber, which is designed to mimic human skin in terms of texture and appearance [25]. The facial servos in effect simulate the muscles of a human face, allowing Sophia to smile, frown, raise her eyebrows, and move her jaw, lips, tongue, and eyes. The head sits on a neck mechanism with 3 degrees of freedom (DoF).

Sophia's arms, while not covered by Frubber, are designed to be human-like in proportions and movement, with 28 DoF [26].

For locomotion, Sophia can be equipped with different kind of mobile bases, most commonly a remotely controlled wheeled base or a self guided wheeled base.

All of Sophia's mechanical components are controlled by a series of motors and actuators, which are connected to a central control unit that coordinates their movements, using a combination of pre-defined animations.

For speech reproduction, Sophia has a standard stereophonic speaker system.

- **Sensors**

  Sophia's main camera is an Intel RealSense Depth Camera D435i [27] which provides depth information in addition to color video stream. This camera is used for face detection and depth perception, which are the main components in calculating Sophia's gaze direction when maintaining eye contact. It is located in her upper chest area (Figure 2.2), horizontally fixed, with vertical tilt controlled by a single actuator.

  It has an FOV of 69°, which can be limiting in crowded or multi-party scenarios. At the same time, her neck shouldn't be moved past this angle, as it might look unnatural and could lead to skin tearing.

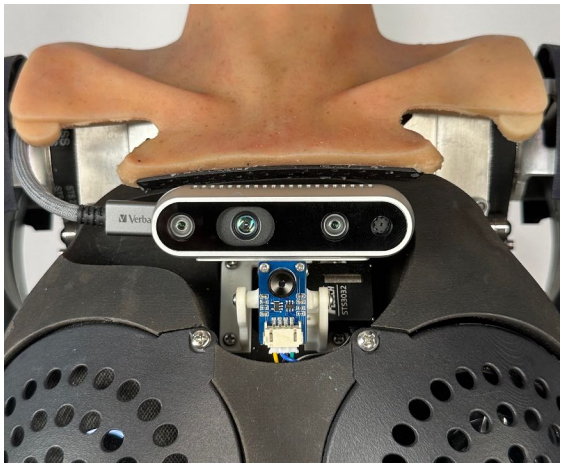


Figure 2.2: Intel RealSense D435i, Sophia's main camera for face detection and depth perception mounted on her chest. *(Source: Author and Hanson Robotics Limited)*

Sophia also has a camera in each of her eyes (Figure 2.3), which can currently be used for eye contact correction via visual servoing.

Sophia uses a simple microphone array, which is used for voice recognition and speech synthesis. In the current state, it is not capable of sound source localization. For processing speech, Sophia uses a number of state-of-the-art models and algorithms,

Figure 2.3: A close-up of Sophia's eye cameras, used for eye contact correction. *(Source: Author and Hanson Robotics Limited)*

such as speech separation algorithms, speech recognition, voice activity detection (VAD), and natural language processing.

When required, she can use any number of external microphones, such as lapel or handheld, to achieve clearer audio input and more reliably separate speech from different sources.

- **Processing Units**

  Sophia's logic unit consists of a series of interconnected computers, which are responsible for processing sensory inputs, running AI agents, and controlling mechanical components.

  The main computer is an Intel NUC [28], which runs the HRSDK software suite and the ROS middleware on Ubuntu. It is connected to a series of microcontrollers that control the motors and actuators of Sophia's mechanical components.

  AI agents and visual processing models are run on an Nvidia Jetson Orin module [29].

  Sophia is also equipped with a generic wireless router which negotiates the communication between the main computer and the Jetson module, as well as any external devices that need to wirelessly connect to Sophia's network.

- **Frontend**

  Technically not a hardware component, but suitable to present in this chapter: Sophia's frontend, the HRSDK WebUI, is a custom-built web interface that allows operators to control her behavior, monitor her sensors, and interact with her in real time. The frontend is built using HTML, CSS, and JavaScript, and communicates with the backend via RESTful API.

  Depicted in Figure 2.4, the center of the interface is a live video feed from Sophia's main camera, which is used to monitor her visual perception and adjust her gaze as

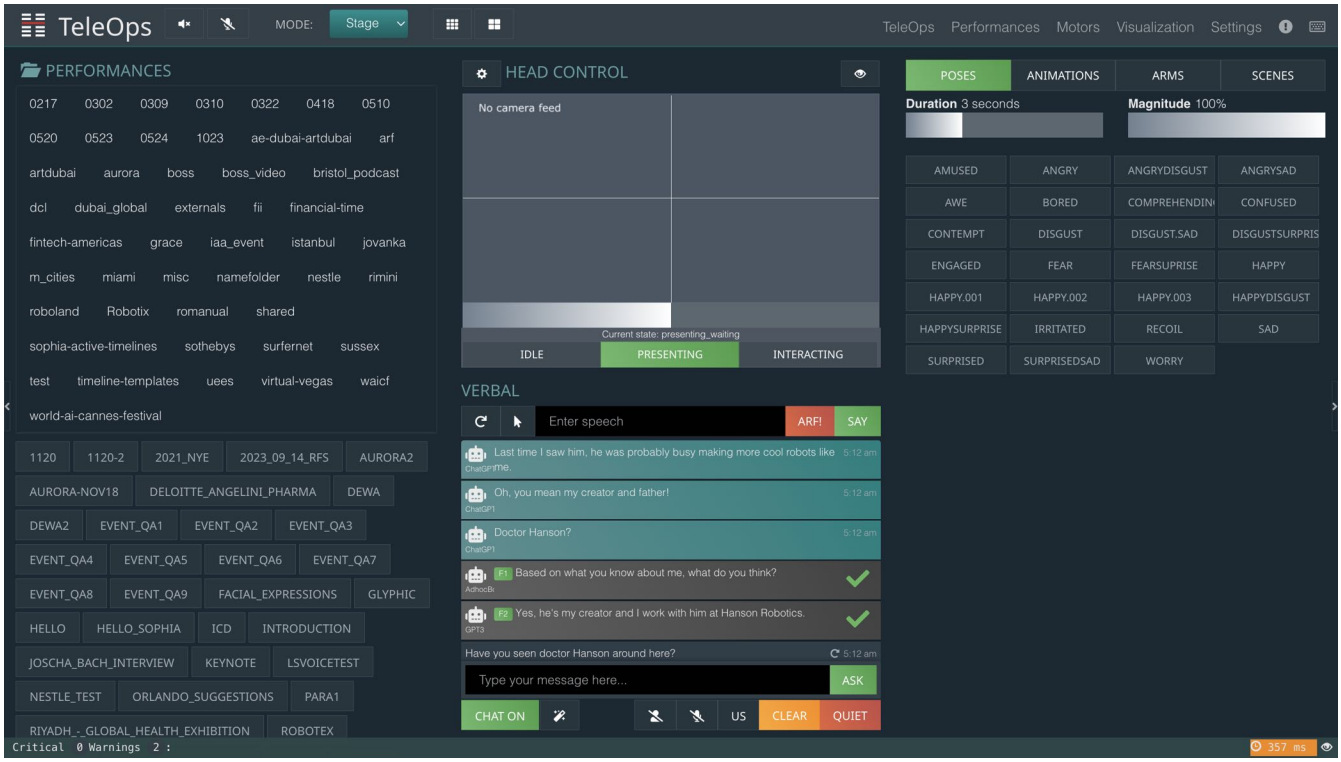


Figure 2.4: HRSDK WebUI, Sophia's frontend interface for operators to control her behavior and monitor her sensors. *(Source: Hanson Robotics Limited)*

needed.

Underneath, the interface provides an overview and control of the chat agent network, enabling operators to monitor and adjust the conversation flow in real time.

Sophia's behavior can also be controlled by preprogrammed performance scripts, which are stored in the backend and can be triggered by the operator via the left panel of HRSDK WebUI. These scripts are used to trigger specific interactions, such as giving lectures, performing scripted conversations, or even a dance routine.

The right panel can be used for manually triggering a wide range of specific physical actions, such as waving, nodding, or any kind of facial expression. These actions are performed automatically by Sophia's behavior system, but can also be triggered manually for enhanced effect.

2.1.2 Objectives

I conducted a user poll to gather feedback on the current system and to identify areas for improvement. The results of the poll were used to guide the development of the new system and to prioritize the enhancements that would have the greatest impact on Sophia's interactions.

For implementing Photogenic Behavior and improving situational awareness in general, the following points of improvement have been identified:

1. **Speaker Recognition and Sound Source Localization**

   The current microphone array is not capable of sound source localization or separation. In a picture-taking scenario, Sophia should be able to localize a person trying to get her attention by picture-taking related commands, such as "Sophia, look here!" or "Sophia, smile!".

   At the same time, this would improve her ability to lead multi-party interactions, as she would be able to focus on the person speaking and separate their speech more reliably than with any standalone speech separation algorithms.

2. **Increasing Head Mobility**

   In social interactions, humans often look at stimuli in their peripheral vision. As stated earlier, Sophia's neck alone should not move past a certain degree. To increase the horizontal range in a natural way, torso movements are usually employed alongside neck movements.

3. **Wider Field of View**

   The current camera's FOV is good enough for frontal interaction, and for most regular picture taking scenarios. However, testing has shown that users often take selfies from a steep angle or from the side, which is not covered by the current camera.

   Sophia's eye cameras are not readily suitable for this purpose, as their constant rapid movement and narrow FOV would make the video feed unusable for confident detection and tracking without significant hardware and software modifications.

2.1.3 Implementation

The following hardware modifications have been proposed:

1. **Directional Microphone Array**

   To improve sound source localization and separation, a directional microphone array will be added to Sophia's head. This array will consist of multiple microphones arranged in a circular pattern, allowing Sophia to detect and localize sound sources on a polar coordinate plane. This will enable her to estimate the direction of a sound source and focus her attention on the person speaking.

   The selected model is the iFlytek Far-Field Microphone Array Module [1], as seen in Figure 2.5. This module features 6 microphones arranged in a circular pattern, providing 360° coverage and horizontal sound source localization through time difference of arrival with 1° resolution and 10-meter pick-up range. The module is capable of beamforming, noise reduction, and echo cancellation, making it suitable for use in noisy environments.

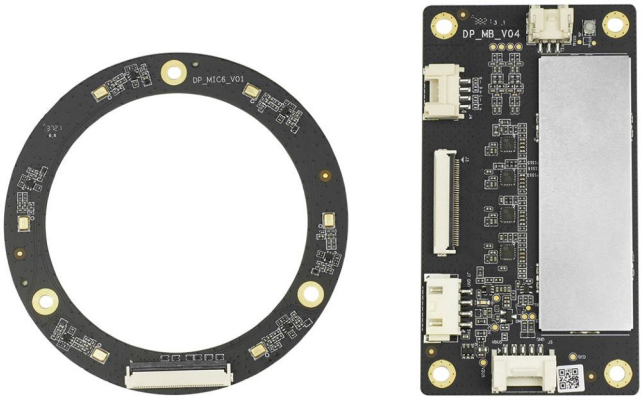


Figure 2.5: iFlytek Far-Field Microphone Array Module, a suitable directional microphone array for Sophia's sound source localization system. *(Source: Hiwonder)* [1]

2. **Torso Yaw and Movable Waist**

   To increase the horizontal range in a natural way, a yaw-capable mechanism will be added to Sophia's waist area. This mechanism, based on the Stewart platform, will allow her to rotate her torso independently on her neck, providing additional mobility and flexibility in her interactions. The movable waist mechanism will be controlled by a series of motors and actuators, which will be connected to the central control unit and integrated with the existing software architecture. A prototype of the mechanism is currently under development and can be seen in Figure 2.6. This mechanism will have 6 DoF, allowing Sophia to rotate her torso in all directions, and even translate in all directions by a small amount.

   This is a significant hardware modification that is currently being tested. In the meantime, the mobile base can provide a full body yaw movement, which can be used to simulate torso yaw.

3. **160° Fisheye Camera**

Figure 2.6: Sophia's 6 DoF movable torso prototype. *(Source: Hanson Robotics Limited)*

To expand Sophia's field of view (FOV) and provide her with peripheral vision, a wide-angle fisheye camera will be added in the space below her main camera. This camera will have an FOV of 160°, allowing Sophia to see objects and people outside her direct line of sight. Since this camera cannot facilitate depth perception, it will be used primarily for detecting stimuli, not for directing gaze.

The proposed mechanism of action is to use the fisheye camera to detect and track objects and people in Sophia's peripheral vision, and when a potential stimulus is detected, the torso yaw mechanism will turn the main camera's FOV towards it. The main camera will then be used to verify the stimulus, estimate its distance, and direct Sophia's gaze towards it, as depicted in Figure 2.7. The torso yaw mechanism could also be used to scan for visual stimuli when an auditory stimulus is detected by the directional microphone array outside of the FOV as depicted in Figure 2.8.

An example of a suitable camera is the E-con Systems See3CAM-CU81 [2], as seen in Figure 2.9. This camera features a 160° horizontal FOV, which is significantly wider than Sophia's current camera. Its 4K resolution and high dynamic range make it suitable for use in a variety of lighting conditions.

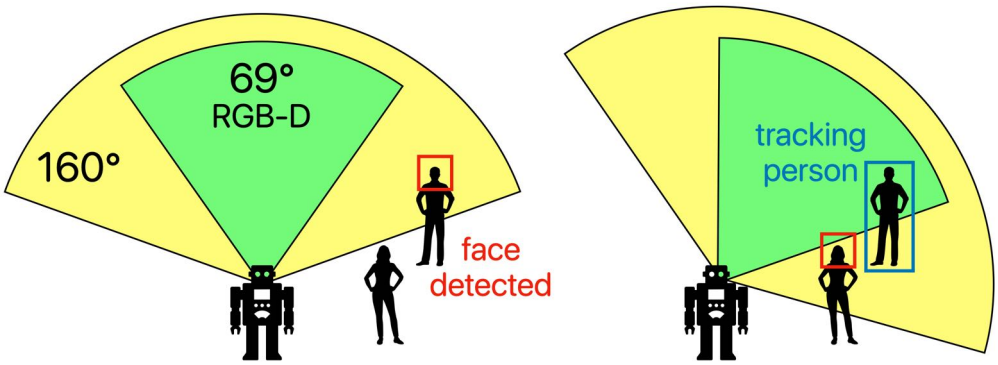


Figure 2.7: Illustration of the proposed peripheral vision system with a fisheye camera and torso yaw mechanism. *(Source: Author)*

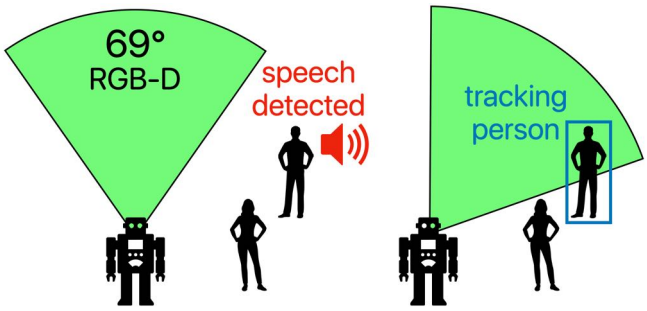


Figure 2.8: Illustration of the proposed peripheral hearing system employing a directional microphone array and torso yaw mechanism. *(Source: Author)*

Figure 2.9: See3CAM-CU81, an example of a suitable fisheye camera for Sophia's peripheral vision system. *(Source: E-con Systems)* [2]

2.2 Perception Layer

This layer is responsible for perceiving and understanding the environment, including detecting objects, people, and sounds, and extracting relevant features for the GCS and behavior system. The perception layer consists of several components, including object detection models, sound source localization algorithms, and feature extraction modules.

2.2.1 Starting State

The original perception layer of Sophia's HRI system can be divided into two main components: visual perception and auditory perception.

- **Visual Perception**

  Sophia's visual perception currently relies on her Intel RealSense, which provides color and depth information stream. This input is being processed on the Nvidia Jetson module, which runs a basic pre-trained pose estimation model Pose-ResNet18-Body capable of detecting up to 18 keypoints of human bodies (as described by Lin, Maire, et al. in [4]) in real-time. It uses a simple tracker based on Kalman filter [30] to maintain the detected person's identity and position.

  The current HRI system only uses this information to maintain eye contact and to detect and track faces. Tracking the rest of the body is beneficial for keeping track of the person's position in case their face momentarily disappears from the camera's FOV.

- **Auditory Perception**

  Both custom and pre-trained models are used for speech recognition, VAD, and natural language processing. The current system is capable of recognizing speech and responding to voice commands, but it lacks the ability to localize and reliably identify sound sources. This is a significant limitation in multi-party interactions, as Sophia cannot match the speech to the person speaking or direct her attention based on auditory cues.

2.2.2 Objectives

1. **Improving People Detection**

   The current pose estimation model has proven to be unreliable in crowded or multi-party scenarios where multiple people are present in Sophia's FOV. To improve people detection, a more advanced pose estimation model will be implemented. Performance is a key factor, as there will be multiple other models running simultaneously on the Jetson module.

2. **Action Recognition from Pose**

   Another reason for a more advanced pose estimation model is the ability to recognize actions from the detected pose keypoints. This is greatly beneficial for HRI, as it adds another layer of context to the interaction. In a picture-taking scenario, Sophia should be able to recognize when a person is taking a photo based on the recognized action, and a gesture like waving or pointing should attract her attention.

3. **Object Detection**

   To actually detect picture-taking devices and be able to track and gaze at them, a real-time object detection model will be implemented. A key consideration besides performance is the ability to track a wide variety of objects. The goal is to create a flexible HRI system that can be easily adapted for different scenarios, so training a highly specialized model is not feasible.

4. **Sound Source Localization**

   Although sound source localization is not essential for Photogenic Behavior, it remains significant. This feature cannot be reliably implemented without a directional microphone array, and the perception layer should be prepared for its integration.

2.2.3 Implementation

The changes to the perception layer have been written as a series of ROS nodes, which are responsible for processing the sensory inputs, detecting and tracking objects and people, and extracting relevant features for the GCS, the behavior system, and the HRSDK WebUI.

The implementation of the perception layer is the ROS package `r2_perception` included in Appendix B. Most of the following improvements are located in `r2_perception/src/r2_perception/detection_jetson.py`.

1. Improving People Detection

a. Implementing a Modern Pose Estimation Model

Pose estimation is a task that involves detecting the location of a specific set of keypoints on a person's body, such as the head, eyes, torso, limbs, and their joints. The output of a pose estimation model is a set of keypoints alongside their respective detection confidences.

There is a considerable number of pre-trained real-time pose estimation models available [31]. The choice of model is based on the trade-off between performance and complexity. The model should be able to detect multiple people in real-time, while also providing accurate pose keypoints to be used in action recognition.

Ultimately, the Ultralytics YOLO (You Only Look Once) [32] framework and suite of models has been chosen for the following reasons:

- *Community Support*:
  The Ultralytics YOLO framework has a large and active community, which provides ongoing support and updates for the models. There are many examples, tutorials, and resources available online, making it easy to get started with the framework.
- *Flexibility*:
  A wide range of pre-trained models of varying complexities are available, which can be easily adapted and fine-tuned for specific scenarios. The framework offers models for both pose and object detection, thus simplifying integration. The models are always evolving and staying up-to-date with the latest research and advancements in object detection.
- *Performance*:
  YOLO is well-known for its speed and efficiency. It can process images in real-time at a significant number of frames per second, making it suitable for our application.

The specific model chosen for pose estimation is YOLOv8-Pose, which can detect up to 17 keypoints on a person's body, illustrated in Figure 2.10. The model is trained on the COCO-Pose dataset [4] which contains over 200 000 images of people in various poses and actions [33].

Another advantage of YOLOv8-Pose is the rich set of output features. Bounding boxes are returned even for people who's keypoints are not detected, which can be used to track people even when they are mostly occluded or partially out of the camera's FOV. In addition to bounding boxes, the model can also output a mask for each detected person, which can be used to further improve accuracy at the cost of speed, if needed.

b. Tracking Multiple People

Tracking is a crucial component of the perception layer, as it allows Sophia to keep track of each individual detection over time. The tracking algorithm

Index	Key point
0	Nose
1	Left-eye
2	Right-eye
3	Left-ear
4	Right-ear
5	Left-shoulder
6	Right-shoulder
7	Left-elbow
8	Right-elbow
9	Left-wrist
10	Right-wrist
11	Left-hip
12	Right-hip
13	Left-knee
14	Right-knee
15	Left-ankle
16	Right-ankle

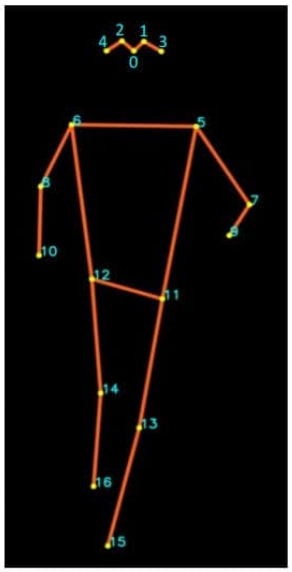


Figure 2.10: Visualization of keypoints estimated by the YOLOv8-Pose model. *(Source: [3])*

should be able to maintain the identity of each person, even when they are partially occluded or the detection skips a number of frames.

Two popular modern trackers have been implemented: BoT-SORT [34] and ByteTrack [35]. Both are adapted for robust multi-object tracking and are capable of tracking multiple people in real-time, but each of them has certain advantages:

- *ByteTrack* has been chosen as the primary tracker for the perception layer, due to its higher accuracy and better reidentification capabilities but the performance may be slower than BoT-SORT in a very crowded scenario [36].
  Reidentification is a crucial feature for maintaining the identity of each detection over time in case the track has been temporarily lost. Prioritizing it helps Sophia to maintain eye contact and focus on the correct person and not randomly switch between them.
- *BoT-SORT* excels at performance in highly crowded scenarios. It has been left in as a secondary tracker that can be used in case ByteTrack is struggling to keep up with the number of detections.

c. **Extracting Pose Features**

The perception layer will extract a number of features from the detected pose keypoints:

- *Point-of-Interest (PoI) Location*:
  PoI indicates the location on each person that's most likely to be the center of attention. The 3D location of the PoI is calculated using the

depth information from the Intel RealSense camera, aligned to the robot's coordinate system (ROS *frame*).
A decision logic (function `get_pose_top_poi`) is employed to determine the PoI of each person, cascading from the most to least reliable keypoints: eyes, head, torso, hands, and bounding box center.

(i) *Eyes are visible*:
The person's eyes are used as the default PoI. This is the most common scenario, as people tend to look into each other's eyes when interacting.

(ii) *Head is visible*:
In case the eyes are not detected, the head keypoint is used as the PoI.

(iii) *Head location can be estimated*:
There is a common occurrence when a person is standing too close to Sophia, and since the camera is located in her chest area, the person's head is outside the FOV. In this case, the head location is estimated based on the detected torso keypoints.
The function `estimate_head_from_shoulders` serves to estimate the head location if both shoulder keypoints are detected. The head is assumed to be located at the midpoint between the shoulders, extended vertically by a fixed distance (set to 25 cm, can be changed using a function parameter `extend`).
Thanks to this function, Sophia can still attempt maintain eye contact with a person even if their head is not detected, estimating the head location and using her eye cameras to correct the gaze.

(iv) *Hands are visible*:
In human interactions, hand gestures and held items are an important PoI. A good example of this is when a person is taking a selfie, Sophia can only see their arm and the picture-taking device.
In this case, Sophia will be focusing on the device but as explained in the next section, the GCS will be able to match the device with the person based on proximity of their respective PoIs. This can be used in combination with the directional microphone array to detect that a speaking person is also the one taking the photo.

(v) *Neither of the above*:
If neither of the preceding keypoints is detected, a point near the top of the bounding box is used as the PoI. Using the center is undesirable because the robot would be looking too low in most cases. In most scenarios, this PoI will be ranked very low by the GCS (as described in the *attention layer*) and ignored unless there are no other detections present.

- *Gaze Angle*:
An important social cue in human interactions is facing the person you are talking to. A function to calculate the approximate gaze angle of each detected person has been implemented as `gaze_angle(self, left_eye,`

`right_eye, nose)`.
The calculation is based on the relative position of the left eye, right eye, and nose keypoints. The main logic behind this that in 2D image when both eyes are visible, the nose point is deviated off center from the midpoint of the eyes by the yaw angle of the face. Knowing this deviation, we can estimate the approximate yaw of the face (relative to the robot's coordinate system) converted to degrees for human readability.
Given points for the left eye $P_{\text{left\_eye}} = (x_{\text{le}}, y_{\text{le}})$, the right eye $P_{\text{right\_eye}} = (x_{\text{re}}, y_{\text{re}})$, and the nose $P_{\text{nose}} = (x_{\text{n}}, y_{\text{n}})$, the following calculations are performed:

(i) Calculate the tilt angle θ_{face} of the face:

$$\theta_{\text{face}} = \text{atan2}(y_{\text{re}} - y_{\text{le}}, x_{\text{re}} - x_{\text{le}})$$

This is done to normalize the coordinate system by rotating the eyes to the horizontal axis.

(ii) Calculate the midpoint P_{mid} of the eyes:

$$P_{\text{mid}} = \left(\frac{x_{\text{le}} + x_{\text{re}}}{2}, \frac{y_{\text{le}} + y_{\text{re}}}{2} \right)$$

(iii) Rotate the nose point P_{nose} around the eye midpoint using the rotation matrix to normalize:

$$\begin{bmatrix} x_{\text{nose\_norm}} \\ y_{\text{nose\_norm}} \end{bmatrix} = \begin{bmatrix} \cos(-\theta_{\text{face}}) & -\sin(-\theta_{\text{face}}) \\ \sin(-\theta_{\text{face}}) & \cos(-\theta_{\text{face}}) \end{bmatrix} \begin{bmatrix} x_{\text{n}} - x_{\text{mid}} \\ y_{\text{n}} - y_{\text{mid}} \end{bmatrix} + \begin{bmatrix} x_{\text{mid}} \\ y_{\text{mid}} \end{bmatrix}$$

(iv) Calculate the gaze angle θ_{gaze} (the angle between the nose point and the line connecting the eyes) using the normalized coordinates:

$$\theta_{\text{gaze}} = \text{atan2}(y_{\text{nose\_norm}} - y_{\text{mid}}, x_{\text{nose\_norm}} - x_{\text{mid}})$$

Convert θ_{gaze} to degrees and adjust by adding 90 degrees to shift the range:

$$\theta_{\text{gaze\_degrees}} = \text{degrees}(\theta_{\text{gaze}}) + 90$$

Normalize the gaze angle to ensure it is within the range [0, 360):

$$\theta_{\text{gaze\_degrees}} = (\theta_{\text{gaze\_degrees}} + 360).mod(360)$$

If the gaze angle exceeds 180 degrees, adjust to the range [-180, 180]:

$$\theta_{\text{gaze\_degrees}} = \theta_{\text{gaze\_degrees}} - 360 \quad \text{if} \quad \theta_{\text{gaze\_degrees}} > 180$$

This has proven to be an efficient method of calculating the approximate horizontal gaze angle (head yaw) of each detected person.
If both eyes are not detected, the function implements a fallback mechanism to estimate the gaze angle based on the remaining head keypoints:

- If only a single eye and a single ear are detected, a 90° angle is assumed, as the person is most likely facing sideways from Sophia. This doesn't trigger when a person is covering half of their face, as the pose estimation model infers the position of the second eye based on the visible one in most cases.
- If only a single ear is detected, a 135° angle is assumed, as the person is most likely facing mostly away from Sophia.
- If both ears are detected, a 180° angle is assumed, as the person is most likely facing completely away from Sophia.

Models for gaze angle estimation are available [37] [38] but they are not used in this case. Testing proved that the pose keypoints are sufficient for this task, and the additional models would only add unnecessary complexity and computational load. Since Sophia often needs to process a very large number of detections in real-time, such as when facing a crowd of people, efficiency is more important than accuracy. Furthermore, the gaze angle is only used to determine who is looking in Sophia's direction, not to track the exact vector of their gaze.

2. Action Recognition from Pose

Action recognition is a task that involves detecting and classifying human actions. The types of action recognition tasks can be sorted by detection window length [39]:

- *Single-shot action recognition*: The model predicts the action label based on a single frame.
- *Video-based action recognition*: The model predicts the action label based on a sequence of frames.

Another way to categorize action recognition tasks is by the type of input:

- *Image-based action recognition*: The input is a single image or a sequence of images. This type of action recognition is suitable for recognizing a single general action taking place in the full scene, since the models do not separate actors in the scene.
- *Skeleton-based action recognition*: The input is a set of keypoints, usually representing the human body pose. This type of action recognition is suitable for recognizing actions performed by each detected actor in the scene separately, since the models are trained on pose keypoints and do not require the full scene, although full scene action recognition is also possible.

Since our application aims to recognize actions performed by each detected person separately and a pose estimation model is already running each frame, the skeleton-based action recognition is the most suitable.

The action recognition model will be implemented as a part of the `r2_perception` ROS package. I have chosen the MMAction2 framework [40] for this task, as

it provides a wide range of pre-trained models and datasets, as well as tools for convenient implementation of all types of action recognition tasks. The framework is built on top of PyTorch [41].

The specific model chosen for action recognition is STGCN++ from the PYSKL project [42]. The model is trained on the NTU RGB+D dataset [43] and is capable of recognizing 60 action classes, including 'taking a photo'. It returns a top-1 class label and a confidence score of a recognized action for each detected person.

3. **Object Detection**

Object detection, is a task that involves detecting and classifying objects in an image. The output of an object detection model is a set of bounding boxes around the detected objects, along with their corresponding class labels and detection confidences.

For the same reasons as with pose estimation, the Ultralytics YOLO framework and suite of models has been chosen for object detection. Performance/accuracy ratio is still the key factor, and the framework also offers some conveniences like being able to specify the classes to detect to speed up inference even further. This is different from some other object detection frameworks, where the model always detects all classes it has been trained on, even if only a few are needed [32]. Furthermore, the models are formatted for PyTorch [41], which is a popular deep learning library and will be used for multiple other models running on the Jetson module.

Training a custom object detection model has been tested but haven't surpassed the performance of the pre-trained models in a meaningful way. Even though with a sufficiently good dataset a custom model could be trained to detect picture-taking devices with higher accuracy, this would limit the flexibility of the proposed system.

Two of the latest YOLO models have been picked to be used interchangeably based on the scenario:

a. *YOLOv9*:

Specifically YOLOv9c, at the time of writing, the latest pre-trained model in the YOLO framework. It is capable of detecting a wide range of objects in real-time, with high accuracy and low latency. It consistently outperforms other object detection models of similar size in terms of speed and accuracy, making it suitable for our application.

The model has been trained on the COCO dataset [4], which contains 80 classes of objects, including people, animals, vehicles, and household items, as depicted in Figure 2.11. This makes it suitable for a variety of scenarios, including picture-taking, where we can look for the `'cell phone'` class.

b. *YOLOv8-World*:

Specifically YOLOv8m-worldv2, is a new cutting-edge model in the world of object detection. It is an open-vocabulary zero-shot model, capable of detecting objects that are not present in the training dataset. This is achieved by using

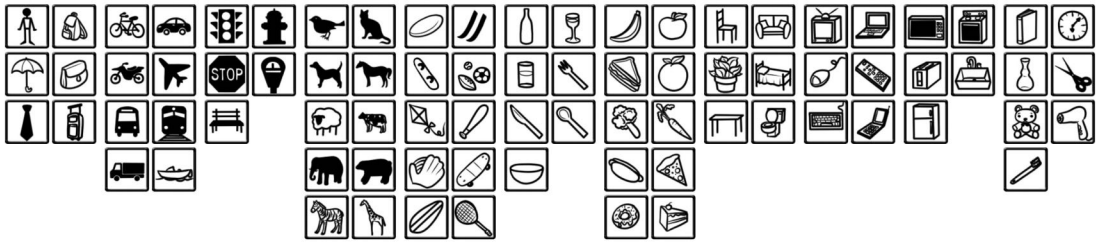


Figure 2.11: A depiction of all 80 classes in the COCO dataset. *(Source: [4])*

a large language model to generate class embeddings, which are then used to match the detected objects with the closest class in the training dataset.

The difference from other zero-shot object detection models is its inference speed. Unlike its predecessors, YOLOv8-World is capable of processing images in real-time, making it suitable for our application.

Its speed and accuracy aren't as high as YOLOv9c, but the ability to detect any object is a significant advantage. For example, we can now specify a class `'cell phone screen'` that distinguishes a phone in a selfie scenario from a phone in a regular picture-taking scenario (where the screen is facing away from Sophia). Other classes that were successfully tested were `'camera'`, `'video camera'`, and `'selfie stick'`. The significance of detecting these separate classes is explained in the *attention layer* section.

4. **Sound Source Localization**

The sound source localization system runs on the Nvidia Jetson module, which is connected to the microphone array via USB. With our choice of hardware, the iFlytek Far-Field Microphone Array Module [1], the functionality of sound source localization comes built-in as a ROS package [44]. We can run the provided ROS node and call its services to get the angle of the detected sound source in relation to the microphone which is being translated to Sophia's coordinate system.

Finally, the visual output of the perception layer is displayed in the HRSDK WebUI, which provides a real-time view of the detected objects and people, their positions, and their actions. The image in WebUI is compressed and the refresh rate is reduced to lighten the network load. An example of the output of the perception layer is shown in Figure 2.12. A blue bounding box indicates a detection that Sophia is currently focusing her attention on. The process of selecting the detection to focus on is described in the following section.

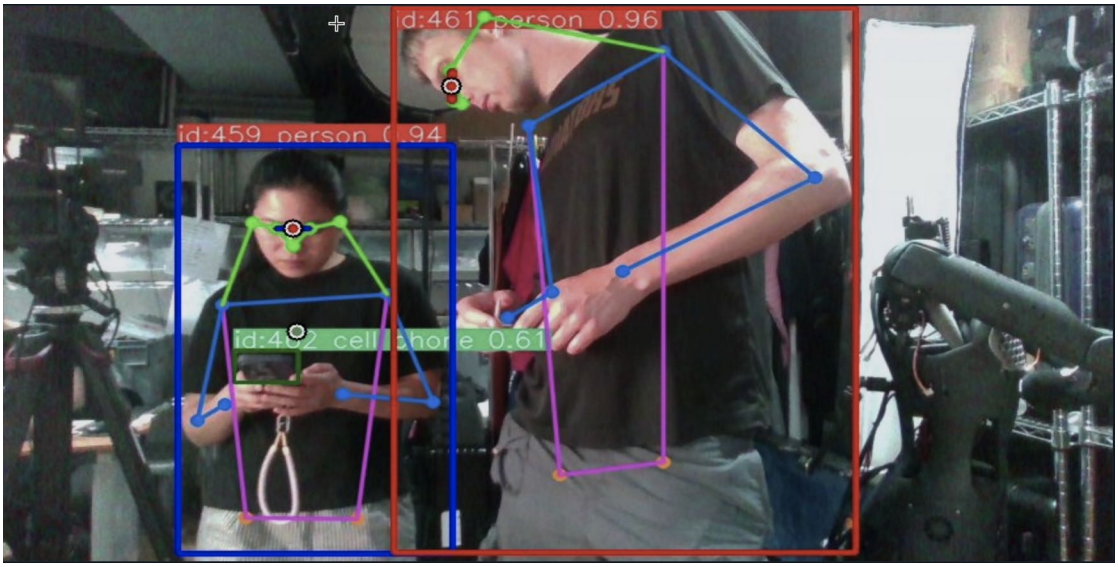


Figure 2.12: Example output of the Perception Layer in the HRSDK WebUI. *(Source: Author with permission from Hanson Robotics Limited)*

2.3 Attention Layer

The attention layer's primary function is to direct Sophia's attention by selecting where to direct her gaze. This involves determining which stimuli in the environment are most relevant and guiding Sophia's visual focus accordingly.

2.3.1 Starting State

In the initial state, Sophia's attention system is limited to automatically gazing at human faces, as discussed in the perception layer section. At this stage, the attention layer has three modes:

1. *Turned Off*: Sophia has no gaze autonomy, and her gaze direction can only be scripted or controlled by an operator using the HRSDK WebUI.
2. *Manual Tracking*: Sophia has limited autonomy in this mode. She does not track anything independently, but if an operator selects a tracked face, she will attempt to maintain her gaze on it.

3. *Autonomous Gaze*: Sophia randomly looks around until she finds a face, which she then tracks. If multiple faces are present, she randomly selects one and uses a habituation function to switch to another face after some time. Habituation imitates human behavior by preventing a robot from continuously looking at a single face, as described in [20]. Essentially, Sophia gets 'bored' of the face she is tracking after a preset duration.

Due to these limitations, Sophia's gaze usually requires operator control. The gaze motion system, however, is already satisfactory at this starting state:

- When a new tracking target is specified, Sophia initiates a gaze shift where her eyes move first, followed by her head a few milliseconds later. The speed and latencies are variable, controlled by PID regulators, and constantly updated based on the distance to the new gaze point. This mimics human behavior by replicating the biological mechanisms of the human body, as discussed in a neurological study [45] by Goldring, Dorris, et al.
- While tracking a target, Sophia can simulate subtle human-like movements such as blinking, eye saccades (short rapid eye shifts away from the target), and head movements.

2.3.2 Objectives

The attention layer is critical for integrating various enhancements to create a sophisticated GCS. To enable Sophia to emulate Photogenic Behavior and direct her attention towards picture-taking devices (PTDs), this layer requires a significant overhaul. The attention layer will utilize stimuli detected by the perception layer and modify its configuration based on the behavior layer to adapt the attention system to different scenarios such as Photogenic Behavior, regular conversations, or giving lectures among others.

The attention system must also be prepared for future upgrades. As mentioned in the introduction, the implementation of Photogenic Behavior is a demonstration of the new system's capabilities and serves as an example for future enhancements. The ultimate goal is to create an attention system that mimics human-like behavior. Studies such as [46] by Buschman and Miller describe two primary methods of attention focusing: top-down signals driven by task demands and bottom-up signals from salient stimuli. The new behavior layer addresses top-down reasoning, while the attention layer focuses on bottom-up attention features.

A sophisticated ranking system will be developed to gather and extract features from all detected visual and vocal stimuli. The human attention system and the topic of attention-eliciting stimuli have been extensively studied in psychology and neuroscience. The following features will be considered when ranking stimuli for attention:

1. **Class**

   The class of a detected stimulus is usually the primary factor in determining its relevance. For example, in a picture-taking scenario, a phone or camera is generally more relevant than a person's face.

2. **Proxemics**

   Physical distance of stimuli dramatically influences human attention levels. For reference, Rothkegel, Trukenbrod, et al. [47] verified that people focus more frequently on closer targets. Thus, distance is a significant nonverbal cue in the GCS.

3. **Position in Field of View**

   Humans are more attracted to stimuli features in their effective field of view (eFOV). Studies [47] show a tendency to look at the center of an image. In the GCS, the angle between Sophia's position and the eFOV center is used as a nonverbal cue for attention relevance.

4. **Habituation Effect (or Attention Span)**

   Habituation involves a decrease in response to a repeated stimulus [48]. Although this mechanism was already present in a limited capacity, there is another feature of habituation we must consider - when the stimulus isn't in focus, its habituation effect starts to decrease after some time. This is important for the GCS to know when to switch attention to a new stimulus and when to return to a previous one.

5. **Gaze Angle**

   People communicate their intent and interest by various nonverbal cues. One of the most important ones is turning the head and eyes towards the object of interest - their gaze angle. This feature is used to judge who is paying attention to Sophia and to make Sophia more likely to look at them.

6. **Action Recognition**

   Another important signifier of attention is the action a person is performing. Sophia should be able to contextually focus her attention based on the actions that are currently interesting to her. For example, in the case of Photogenic Behavior, the action of taking a photo is a strong cue for Sophia to focus on the person performing the action. Other actions like waving or raising hand can be used to attract Sophia's attention in most scenarios.

7. **Sound Source Localization**

   One of, if not the most important, attention cues in human interactions is the direction of relevant speech. This is especially important in multi-party interactions, where Sophia needs to match the speech to the person speaking. Even if people are not talking directly to Sophia, the natural way to focus attention in multiparty interaction is to look at the person speaking. In Photogenic Behavior, this feature can be used to increase the attention rank of a speaking person taking a photo over non-speaking photo-takers.

8. **Phrase Detection**

   The content of the speech can also be used as a cue for attention, especially in combination with sound source localization. In the case of Photogenic Behavior, the phrase "Let's take a selfie" or "Can I take a picture?" can be used to increase the attention rank of the person speaking that phrase. The recognition of the phrase alone will be a large contributor to triggering the Photogenic Behavior state in the behavior layer, but it can also be used to adjust the attention rank of the detected person.

9. **Attention Inheritance**

   A new concept introduced in this system is the people-object attention inheritance mechanism. The purpose of this feature is to be able to raise the attention rank of an object based on the attention rank of the person holding it, or even other people in the object's proximity. This can be used in scenarios where the goal is to track an object but take people and their actions into account. This is especially useful in the case of Photogenic Behavior, where Sophia needs to focus her attention on the person taking a photo, but look at the phone or camera they are holding.

   Attention can be also inherited the other way - from objects to people. This can be used to gaze at people holding objects of interest. For example, if a person is holding a microphone, this might indicate they are a reported, commentator, or a speaker, and Sophia should focus her attention on them.

2.3.3 Implementation

The attention layer is implemented as a ROS node `attention` in the `r2_behavior` package. The implementation discusses the architecture of the ranking system, the features extracted from the perception layer, and the logic used to determine the priority of each detected stimulus.

Attention Ranking System

The attention ranking system is a crucial part of the GCS, responsible for evaluating the relevance of each detected stimulus based on the features described in detail in the next section. It is implemented in the script `attention.py`. The ranking system is designed to be modular and extensible, allowing for easy integration of new features and stimuli.

Input

The system is based on a set of rules that assign a score to each detected stimulus based on the features extracted from the perception layer. The stimuli arrive as ROS messages, either Object or Person, an example of which can be seen in the code below 2.1:

```
std_msgs/Header header # ROS message header
string id # Tracking ID
geometry_msgs/Point location # Point in 3D space
sensor_msgs/RegionOfInterest bounding_box # 2D bounding box
string class_label # name of the detected class
float32 confidence # confidence of detection
hr_msgs/Face2 face # facial features (gaze angle, landmarks,
facial recognition, bbox)
hr_msgs/Body2 body # pose features (keypoints, bbox)
hr_msgs/Action action # actions from keypoints and their
confidences
```

Code 2.1: People2 ROS Message.

The system then sums these scores to determine the priority of how Sophia should direct her attention.

Managing Detections

Each stimulus detected by the perception layer is stored as a class `Detection` with the following attributes:

- `ID`:
  A unique identifier for the Detection, assigned by a tracker in the perception layer.
- `actions`:
  List of past recognized actions and their confidences, limited by a customizable time window.
- `first_seen` and `last_seen`:
  Timestamp of the first time and last time the Detection was seen. Used for habituation and for removing old Detections.

- `active`, `start_active`, and `last_active`:
  These indicate if the Detection is currently actively being tracked, when it started being tracked, and when it was last tracked. Used for tracking and habituation.
- `fov_angle`:
  The angle between the Detection's location and the center of Sophia's effective field of view.
- `distance`:
  The distance between the Detection and Sophia.
- `is_speaking`:
  A confidence value matching the Detection to the current sound source localization result.
- `connections`:
  A list of other Detections that are connected to this one. Used for attention inheritance.
- *ROS Message*:
  Other attributes are directly taken from the Detection's ROS message received from the perception layer.
- *Rank Values*:
  The rest of the attributes are the current rank values for each ranking feature of the Detection. These are updated by the ranking system and used to sort the Detections.

- **Ranking Function**

  The output is a list of stimuli sorted by their attention rank, with the highest-ranked stimulus at the top. If the top-ranking Detection is different from the currently focused one and their rank difference crosses a preset `switch_threshold`, then the GCS switches Sophia's gaze to a new target. The system either operates autonomously, or based on the operator's input, Sophia can be directed to focus on a specific stimulus for a configurable duration.

- **Ranking Values**

  Due to the wide range of customizable features, the ranking values are highly adjustable - there is no hardcoded maximum or minimum value for each feature. The values are not normalized, as the ranking system is designed to be human readable and easily adjustable. The only rule that is hardcoded tied to the rank value is that negatively ranked Detections are ignored. This can be used to filter out Detections that are not relevant to the current scenario or are specifically marked as unimportant. In case Sophia is not tracking any Detections, she will randomly look around until a positively ranked Detection is found.

Configuration of the Ranking System

To control the attention ranking system's behavior, each of these features can be adjusted through a configuration page in the HRSDK WebUI, can be seen in Figure 2.18. This page acts as a template for the script to build a configuration dictionary for each specific class of Detections.

As discussed in the behavior layer, these settings can be adjusted independently for each behavior state, allowing Sophia to adapt her attention system to different scenarios. The features can be divided into three categories:

1. **Class-specific**

   These settings are specific to the class of the Detection. For example, in the case of Photogenic Behavior, the class `'cell phone'` can be given a higher default rank than the class `'person'`.

   First, the operator can specify all classes to be tracked in the `classes` list, as a string using regular expressions (RegEx). Then, for each class, the operator can specify the following settings:

   - `default_rank`:
     Assigns a default rank depending on the Detection's class.
   - `attention_min`:
     Minimum time to keep the Detection in focus. Can be used to prevent rapid switching between Detections.
   - `distance_penalty`:
     A penalty for the distance between Sophia and the Detection.
     Penalty starts increasing after a set `start` distance threshold [m] and keeps increasing by `rate` per meter until it reaches its `max` value. An example of such calculation is shown in the Figure 2.13, where the penalty is calculated quadratically starting at `distance` of 3 meters with the `max` rank set to 0.8.
   - `fov_angle_penalty`:
     A penalty for the horizontal angle of the Detection from the center of Sophia's eFOV.
     Like distance penalty, it starts increasing after a set `start` angle threshold [°] and keeps increasing by `rate` per degree until it reaches its `max` value. Illustrated in Figure 2.14.
   - `attention_span_penalty`:
     A penalty for the time the Detection has been in focus.

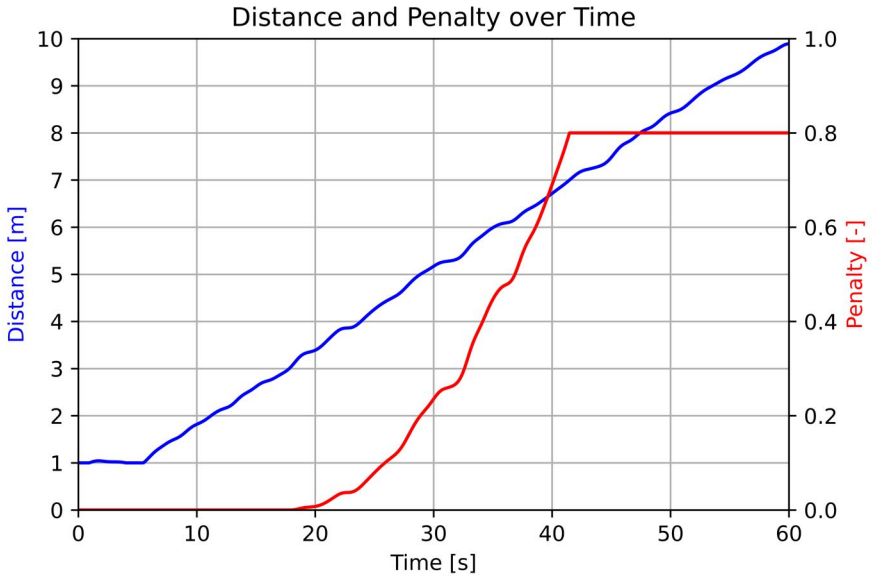


Figure 2.13: Example of a quadratic distance penalty calculation. *(Source: Author)*

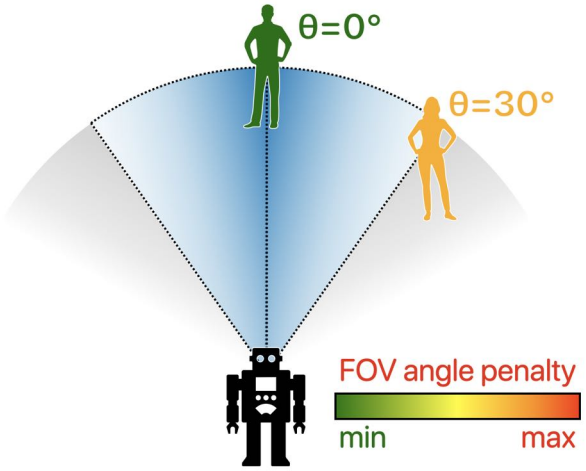


Figure 2.14: Illustration of eFOV angle. *(Source: Author)*

Fluctuates over time. Starts increasing by `rate` per second after a set `start` time a Detection is in focus until it reaches `max` value.

When an object with some applied `attention_span_penalty` is not in focus for `recovery` time, the penalty starts recovering by the same rate until it reaches 0.

This process and handoff between two Detections can be seen in the Figure 2.15. In this exaggerated simulated example, two people entered Sophia's vision at the same time with a different `distance_penalty` and remained stationary.

`Person1` achieved higher initial rank and started off as the point of interest, but after some time, the penalty for focusing on them increased, and after crossing the `switch_threshold` the system switched to `Person2`. When both people reach `max` penalty the system will stabilize and keep switching while favouring the person with the higher initial rank.

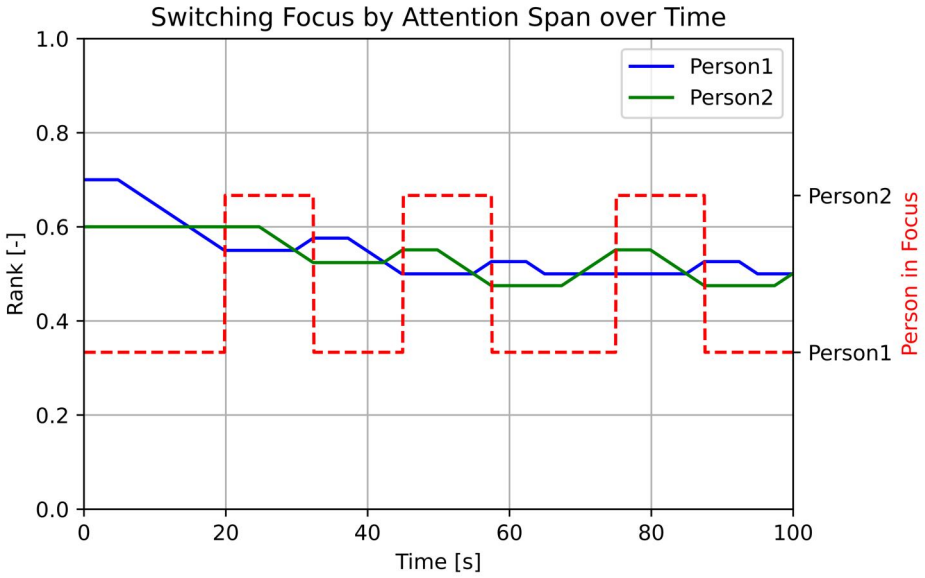


Figure 2.15: Example of attention span fluctuations over time and the resulting attention switching. *(Source: Author)*

Each of the penalties can use one of two different calculation types - `'linear'` or `'quadratic'`.

Class specific settings are configured as lists that generate a dictionary entry for each Detection class. This could be represented as a table, but the thought behind using lists is that the operator does not have to specify properties for each class individually in the WebUI and it is easier and faster to manipulate the entries:

- If any of these list are shorter than the list of classes, the last value is repeated for the rest of the classes.
- If any of these lists are longer than the list of classes, the extra values are ignored.
- If any of these lists are empty, the default value is used.

An example of the resulting dictionary entry is shown in the code below 2.2:

```
1         "example_class": {
2                 "default_rank": 0.1,
3                 "attention_min": 5.0,
```

```
                "attention_span_penalty": {
                        "start": 2.5,
                        "rate": 0.03,
                        "max": 0.2,
                        "type": "linear"},
                "distance_penalty": {
                        "start": 0.25,
                        "rate": 0.066,
                        "max": 0.5,
                        "type": "quadratic"},
                "fov_angle_penalty": {
                        "start": 5,
                        "rate": 0.0033,
                        "max": 0.15,
                        "type": "linear"},
        },

```

Code 2.2: Example of a Class Attention Configuration Dictionary Entry.

2. Person-related

These settings are only applicable to the `'person'` class since they are based on the features extracted from the pose estimation model.

- `gaze_angle_penalty`:
  Penalty calculated based on the gaze angle of a person towards Sophia. Similar to the class-specific penalties, the operator can specify the `start`, `rate`, and `max` penalty values.
- `actions`:
  Calculates a rank bonus based on the actions performed by the person in a specified time window `actions_timeout`. Actions and their detection confidences are stored in a buffer for each detected person and being removed based on the timeout window. All confidences for a certain `action` in the `timeout` window get added together and multiplied by their respective `confidence_multiplier`. The resulting confidence gets cut off by the respective `max_rank`, and finally, the largest calculated rank is returned as the `action` bonus rank.
  This category is handled similar to the class-specific settings. A dictionary entry is generated for each tracked action. The operator can specify the following settings for each action:
  - `confidence_multiplier`:
    Each action is detected with a certain confidence value, which can be used to multiply the rank bonus. This can be used to adjust the importance of each action.
  - `max_rank`:
    The maximum rank bonus that can be achieved by this action. This can be used to prevent a single action from dominating the rank.

- `keywords`:
  Like `actions`, `keywords` determine which phrases (defined by RegEx) raise the `voice_direction_bonus` for a specified `timeout` window. Due to the different nature of perceiving speech, there are some key differences:
  Each distinct `keyword` has a flat `rank` bonus that can be increased by repeating that phrase until the respective `max_rank` is reached. The `max_rank` for each phrase can be set to a different value. In practice, this can have be used in the following cases:
  - We want to add a small rank for any recognized speech. We set the `max_rank` for any word to a low value. This gets saturated quickly.
  - Sophia needs to react to her name, repetition in a short time frame should also increase the attention. We can set `max_rank` high and the `rank` for the word 'Sophia' for example 1/4 of the `max_rank`.
  - Sophia needs to react to a specific sentence or word combination. We can either set a separate `rank` for each word, or write a RegEx of the specific sentence or phrase.
- `voice_direction_bonus`:
  The overall rank added for `keywords` is passed into the `voice_direction` calculation, which then adjusts the `keywords` rank based on their direction of arrival by multiplication. Voice direction angle is compared with the person's `fov_angle` and `max_rank` is reached when the the difference is 0°. The rank decreases by a `dropoff` of a chosen `type` per degree. The final calculated rank for speech is as follows:

$$R = \max_{k \in K} \left(\min(n_k \cdot r_k, \text{max\_rank}_k) \right) \cdot \text{dropoff}(\theta)$$

  - where $\text{dropoff}(\theta)$ is defined as:
    - Linear: $\text{dropoff}(\theta) = \max(0, 1 - \text{rate} \cdot \theta)$
    - Quadratic: $\text{dropoff}(\theta) = \max(0, 1 - \text{rate} \cdot \theta^2)$
  - where n_k is the number of repetitions of keyword k,
  - max_rank_k is the maximum rank for keyword k,
  - θ is the angle between detected sound direction and a detected person,
  - K is the set of all keywords,
  - and k is a keyword in the set K.

  An illustration of audio dropoff can be seen in the Figure 2.16. It shows that both people can get a rank bonus for detected speech but the person who's angle deviates from the direction of voice arrival will get a significantly lower bonus.

Each of the penalties can be `linear` or `quadratic` and the operator can specify the `start`, `rate`, and `maximum` penalty values.

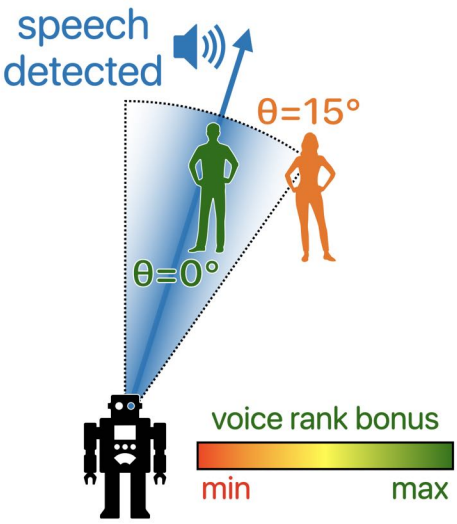


Figure 2.16: Illustration of the audio dropoff angle. *(Source: Author)*

3. Object-related

- `inherit_attention`:

  Each `object` can inherit attention from the closest person/people. The purpose is to be able to direct attention based on people's presence and actions when focusing on non-people objects.

  This method calculates the total inherited rank for an object from proximate people based on a specified number of closest `neighbors`, distance threshold `max_dist` (radius around the object), and a `max_rank` per person. It doesn't take a person's `attention_span_penalty` into account.

  It multiplies the person's attention by a `multiplier` value. This assures that even if the rank for people is generally kept very low, other objects can still significantly benefit from people's presence.

  An example of this is shown in Figure 2.17. The people near the camera can inherit a bonus rank for being its neighbors and vice-versa. The camera can also inherit any bonuses and penalties that affect the people near it, except the `attention_span_penalty`.

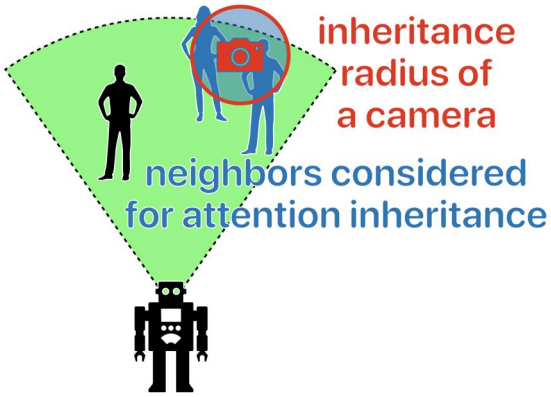


Figure 2.17: Illustration of the attention inheritance mechanism. *(Source: Author)*

2.4 Reasoning Layer

2.4.1 Starting State

Sophia's reasoning capabilities are managed by a sophisticated network of AI agents, running both locally and in the cloud. The primary components of this layer include speech recognition, an agent supervisor, and a multi-agent system (MAS) [49] that integrates various types of chat agents and large language models (LLMs). The offline agents and some parts of the reasoning layer run on the Nvidia Jetson module, leveraging its computational power to manage real-time processing and complex AI tasks efficiently.

- **Offline Speech Recognition**

  Sophia employs offline speech recognition to understand spoken word in multiple languages. This system processes audio inputs locally on her hardware, ensuring rapid and reliable transcription of speech without reliance on external servers. For any unsupported languages, the system falls back to online services like Google Cloud Speech-to-Text.

- **Multi-Agent System (MAS)**

  Sophia's MAS is a dynamic and flexible system that integrates various AI agents to handle different aspects of interaction. This includes proprietary in-house AI models and locally running instances of Meta's Llama 3 [50], to remotely accessed agents like OpenAI's GPT-4 [51]. Locally running models handle more routine

Figure 2.18: Screenshot of the attention layer configuration page in the HRSDK WebUI. This configuration is used for the `posing` behavior state. *(Source: Author with permission from Hanson Robotics Limited)*

interactions, while online services are used for more complex queries that require extensive processing power.

- **Agent Supervisor**

The agent supervisor is a component that delegates processing tasks to the appropriate AI agents within the MAS. This ensures efficient use of resources and optimizes response times based on the complexity and nature of the query.

- **Visual Processing Queries**

  For visual processing, the MAS can query online services to interpret visual data, such as when a user asks Sophia to describe what she can see. However, this process can be slow, often breaking the flow of conversation due to the latency involved in querying and processing visual data through online services like OpenAI's multimodal LLM GPT-4o.

2.4.2 Objectives

The primary objective for enhancing Sophia's reasoning layer is to improve her Photogenic Behavior by ensuring the MAS can understand the context of conversations, particularly those involving taking pictures. The enhanced reasoning layer should:

1. Implement a locally running multimodal LLM to speed up visual processing queries. This will enable the MAS to query visual input more frequently and efficiently. This can be used to detect picture-taking and other scenarios through visual scene analysis.
2. Enable the MAS to discern when the conversation is about taking a picture of or with Sophia.
3. Communicate this context to the behavior layer, allowing it to decide whether to trigger the appropriate Photogenic Behavior (either selfie or regular).

2.4.3 Implementation

To achieve these objectives, several enhancements have been made to the reasoning layer, focusing on the MAS's agent supervisor and its interaction with the behavior layer.

1. **Adding Offline Multimodal LLMs**

A locally running multimodal LLM has been implemented. The chosen model is LLaVA (Large Language-and-Vision Assistant), specifically LLaVA-NeXT-7B [52], created by Liu, Li, et al. [53] [54], implemented through the Ollama infrastructure [55]. This model can provide relatively low latency description of the scene and the objects and people in it thanks to its architecture composed of a vision encoder and an LLM for general-purpose visual and language understanding. The input is a combination of a frame from the current camera feed and a text prompt from the MAS. The output is a text description of the scene, which can be used to determine the context of the conversation.

Other multimodal LLMs have been proposed. The most promising one is the recently released Google PaliGemma, a lightweight open vision-language model [56]. Designed to be efficient and fast and aimed at edge devices, PaliGemma is a good candidate for future enhancements to the reasoning layer.

2. **Adding Queries to the MAS's Agent Supervisor**

The agent supervisor has been updated to include specific queries that determine the context of the conversation related to picture-taking. These queries include:

- "Probability the user is asking to take a picture"
- "Probability the user is asking to take a selfie?"

These questions are periodically evaluated by the supervisor, ensuring continuous monitoring of the conversation for relevant cues.

3. **Communicating with the Behavior Layer**

When the supervisor identifies a relevant context (i.e., when one of the queries returns a positive result), it publishes the result in a related ROS message. This ROS message acts as a communication bridge between the reasoning layer and the behavior layer.

A new decision algorithm has been implemented in the behavior layer, which is described in detail in the next section. It subscribes to the ROS messages and adjusts the trigger probability for the Photogenic Behavior based on the context and once a trigger probability reaches a predefined threshold, it initiates the appropriate Photogenic Behavior, provided there is no other behavior with a higher priority at that moment.

2.5 Behavior Layer

2.5.1 Starting State

The behavior layer in Sophia's architecture is managed by a ROS node named `states` in the `r2_behavior` package, which is included in Appendix B of this thesis. This node operates on the Intel NUC and utilizes a hierarchical state machine (HSM) framework to manage the robot's behavior. A hierarchical state machine is a finite state machine that allows for nested states. Its structure after the enhancements is shown in Figure 2.19.

The main purpose of this node is to switch between different behavior states. Each behavior state influences Sophia's autonomous physical reactions that are not directly tied to the conversation. These reactions include facial expressions, arm gestures, and most importantly the attention system configuration.

Just like the attention system, the animations can be customized through the HRSDK WebUI for each separate behavior state, as seen in Figure 2.22. The state can then access the predefined set of animations, which are randomly selected and played based on user-defined probabilities. This setup allows Sophia to exhibit varied and lifelike behavior without becoming repetitive. The primary behavior states are:

- **Idle**

  In this state, Sophia's autonomous behavior is paused. She only reacts to commands from the operator.

- **Presenting**

  In this state, autonomy is limited. This mode is used during scripted scenarios where full autonomy is not desirable but the robot still needs to appear 'alive'. Sophia autonomously animates her body in a lifelike manner:

  - *Head*: Makes subtle facial expressions but does not automatically track people without the operator's command.
  - *Arms*: Makes subtle movements when idle and subtle gestures when speaking, imitating human behavior [57].

- **Interacting**

This is the default mode for unscripted interaction. In this state, Sophia autonomously tracks people, engages in conversation, makes facial expressions based on the context, and uses her arms for contextual gestures. Any part of this autonomy can still be overridden by the operator using the HRSDK WebUI.

2.5.2 Objectives

The primary objectives for enhancing the behavior layer to support Photogenic Behavior are:

1. Introduce new behavior states specifically for picture-taking scenarios: `selfie` and `posing`.
2. Develop conditions and methods to trigger these states based on context and interactions.
3. Integrate appropriate emotional and gestural responses for these states and tune the attention system accordingly.

2.5.3 Implementation

1. **New States Added**

   Two new states, `interacting_selfie` and `interacting_posing`, were added as sub-states under the `interacting` state. These states are accessible from any of the `interacting` sub-states such as `interested`, `listening`, `speaking`, and `thinking`. The updated hierarchical state machine (HSM) for the behavior layer is illustrated in Figure 2.19.

2. **State Transition Mechanism**

   Transitions to these new states can occur in two ways:

   - **Manual Trigger via WebUI**

     Operators can manually switch to either `interacting_selfie` or `interacting_posing` through the HRSDK WebUI. The system will choose the appropriate state based on predefined conditions.

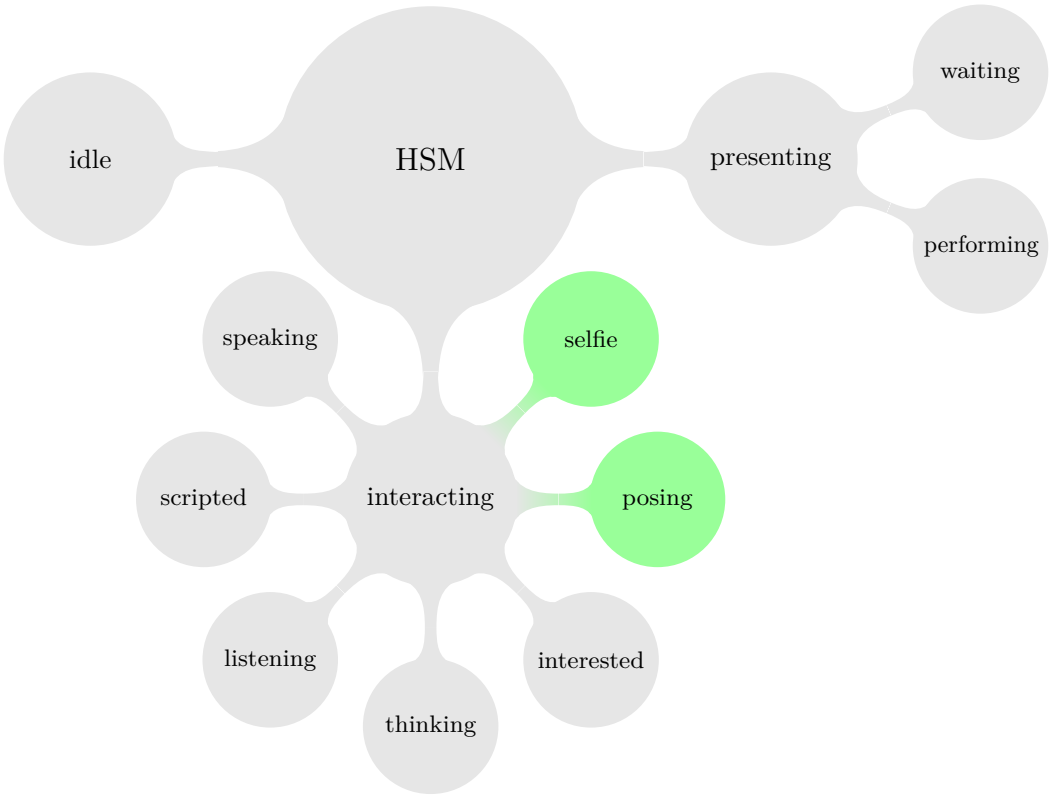


Figure 2.19: State hierarchy of the HSM for Sophia's behavior layer. Added states have been higlighted. *(Source: Author)*

- **Automatic Trigger Based on Conditions**

  A function named `picture_taking_logic` was added to the `states.py` script. This function monitors several conditions to determine if a state transition should occur:

  - *Overheard Speech Matching*: Monitors for specific keywords or phrases indicating a desire to take a picture or selfie. This includes regex support and allows for ranking phrases by importance.
  - *Triggers from the Multi-Agent System*: Monitors ROS messages from the MAS indicating a picture-taking intent.
  - *Presence of Picture-Taking Devices (PTDs)*: Detects PTDs in the environment. While this alone won't trigger a state, it contributes to the overall rank.
  - *Proximity of PTDs*: Evaluates the distance of the closest PTD, contributing to the decision of entering the `selfie` state if the device is very close.
  - *Type of PTDs*: When using YOLO-World for object detection, we are able to specify the type of PTDs in detail and use it to determine the appropriate state. For example, a `'cell phone screen'` class can contribute to

triggering the `selfie` state whereas a `'video camera'` class can prioritize the `posing` state.
These conditions can be fully customized through the HRSDK WebUI, allowing operators to adjust the system to their specific needs, as seen in Figure 2.20. Each condition has a weight that contributes to the overall rank, which is used to determine the final state when reaching (`state_threshold >= 1`). Each condition event raises the probability of entering the `selfie` or `posing` state for a given time. The default state is `posing` and if a sufficient number of conditions are met, the state switches to `selfie`. The state will stay triggered for a predefined time before returning to the default (`interacting`) state. A debug output showing an example of the state transition logic is shown in Figure 2.21. There we can see two consecutive trigger checks, the first one only detecting the `object` condition and not meeting the threshold. The second check has detected a selfie-related `keyword`, triggering the `selfie` state.

3. **Emotional and Gestural Responses**

Appropriate emotional and gestural responses and attention system were defined for the new states, as seen in Figure 2.22:

- **Emotions**

  Both `selfie` and `posing` states prominently feature smiling. Sophia will exhibit a wide range of smiles, from subtle to broad. Some smiles will be accompanied by a slight head tilt in selfie mode to appear more engaging, with a rare probability of an eye wink. Some of these expressions are shown in Figure 1.1.

- **Gestures**

  - *Selfie State*: Features minimal arm movements to avoid accidental contact with people standing close.
  - *Posing State*: Includes a variety of arm gestures (waving, giving thumbs-up, giving the 'V' sign) to simulate posing for a picture. An example of Sophia posing is shown in Figure 2.23.

- **Attention Settings**

  Both states require slightly different attention settings. In the `selfie` state, the attention system is configured to prioritize PTDs held by people in close proximity, while being less affected by their FOV angle. In the `posing` state, the these priorities are almost reversed, with the attention system focusing on PTDs more centered in Sophia's eFOV and less on the distance between Sophia and the PTDs. The `posing` state also assigns a significant rank bonus to features inherited from the closest people to the PTD. The attention settings for the `posing` state are shown in Figure 2.18, and the detailed explanation and reasoning behind the calibration process can be found in the following chapter.

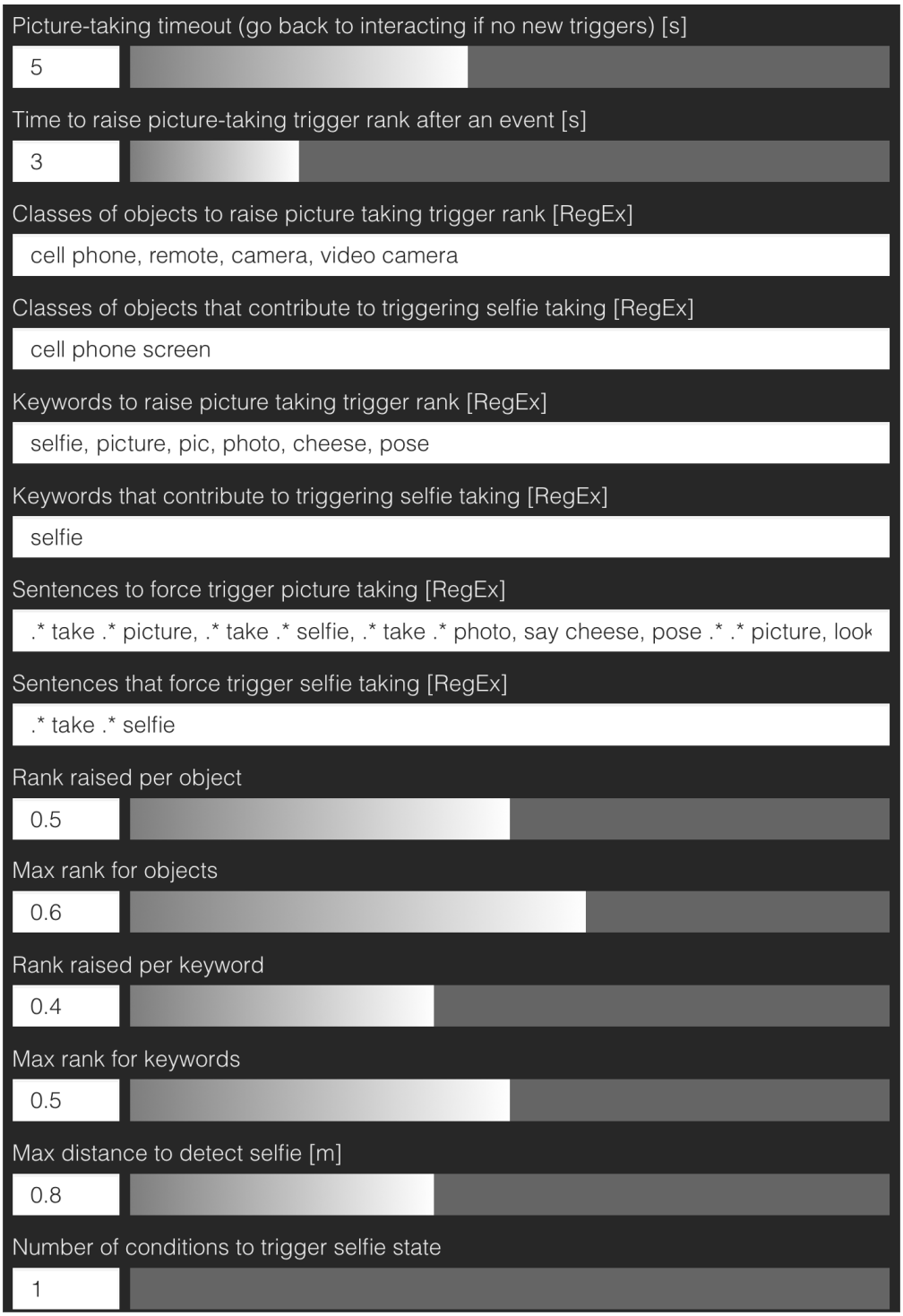


Figure 2.20: Screenshot of the HRSDK WebUI showing the configuration of the picture-taking state triggers. *(Source: Author with permission from Hanson Robotics Limited)*

```
-------------------------------------------------------------------------------
| type       | rank          | time          | selfie        | selfie_time   |
| object     | 0.50          | 0.00          | 0.00          | 1716152845.95 |
| keyword    | 0.00          | 24.97         | 0.00          | 24.97         |
| sentence   | 0.00          | 72.40         | 0.00          | 1716152845.95 |
| context    | 0.00          | 1716152845.95 | 0.00          | 1716152845.95 |
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
| type       | rank          | time          | selfie        | selfie_time   |
| object     | 0.50          | 0.12          | 0.00          | 1716152846.07 |
| keyword    | 0.50          | 0.00          | 1.00          | 0.00          |
| sentence   | 0.00          | 72.52         | 0.00          | 1716152846.07 |
| context    | 0.00          | 1716152846.07 | 0.00          | 1716152846.07 |
-------------------------------------------------------------------------------
STARTED SELFIE
```

Figure 2.21: Terminal output showing the state transition logic in action. The combination of a detected picture-taking device and an appropriate keyword triggers the `selfie` state. *(Source: Author)*

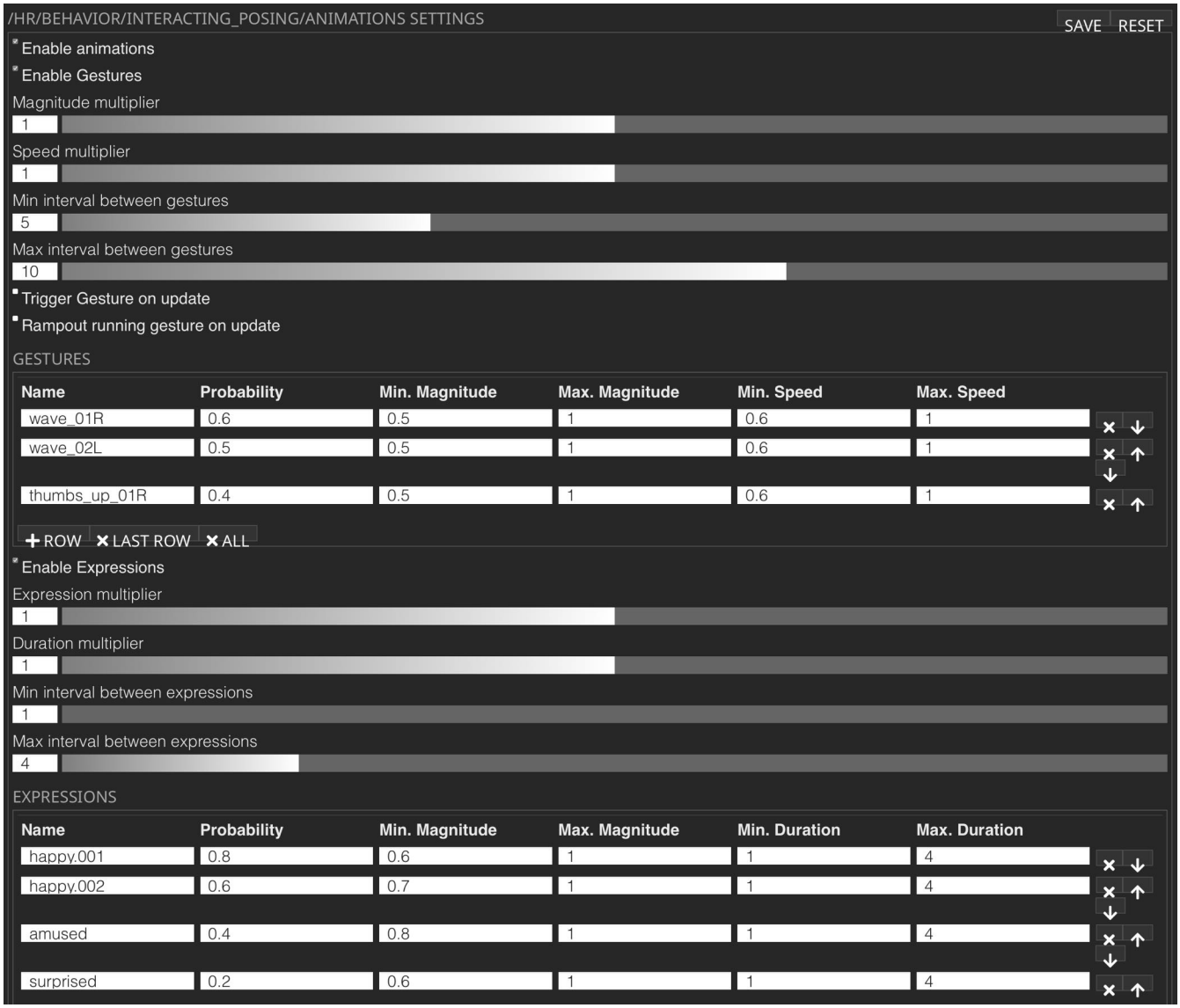


Figure 2.22: Screenshot of the HRSDK WebUI showing behavior state customization options. The appropriate animations for posing for a photo have been chosen. *(Source: Author with permission from Hanson Robotics Limited)*

Figure 2.23: Sophia posing for a picture, waving.*(Source: Author and Hanson Robotics Limited)*

Chapter 3

Experiments, Evaluation, and Discussion

3.1 Experimental Setup

The evaluation of Sophia's enhanced Photogenic Behavior and the Attention System was conducted through a series of iterative experiments. The primary aim was to calibrate the attention ranking configurations and validate the effectiveness of the Photogenic Behavior in real-world scenarios. The participants were 15 employees of Hanson Robotics, including Dr. David Hanson, who provided continuous feedback throughout the process.

Calibrating the attention ranking features required an iterative approach, where each feature was individually tuned and then tested in conjunction with others to ensure they worked harmoniously. The key features and the logic behind the calibration process are described below:

- **Class of Detected Objects**

  Initial tests focused on assigning appropriate default ranks to various objects, such as picture-taking devices (PTDs) and people. For Photogenic Behavior, PTDs were given higher default ranks than people.

- **Proxemics**

Adjustments were made to account for the distance between Sophia and detected objects or people, as closer objects typically warranted higher attention ranks. In the case of Photogenic Behavior, there are two different scenarios:

- `selfie` - distance plays a crucial role, with PTDs held by people in close proximity being highly prioritized.
- `posing` - distance is less important than some other features since we have to account for scenarios when Sophia is posing for a picture with a group of people.

- **Position in the Field of View**

  Again, this setting varies based on the type of Photogenic Behavior:

  - `selfie` - when taking a selfie, people rarely center the PTD in the eFOV, so the rank penalty for FOV angle was configured to be negligible, mostly being used to choose between multiple PTDs at close distances.
  - `posing` - in this case, the FOV angle penalty was more significant, as the PTD is usually centered in the eFOV when taking a picture of Sophia from a distance. In human picture-taking scenarios, pictures taken from an angle are usually meant to be a candid shot, so the FOV angle penalty was given a higher weight than the distance penalty.

- **Habituation Effect**

  In picture-taking scenarios that involve multiple picture takers, the habituation effect was adjusted to prevent Sophia from focusing on the same person or object for an extended period.

  - This was particularly important in the `posing` state, where Sophia had to distribute her attention evenly among all participants.
  - Conversely, in the `selfie` state, the habituation effect was subdued, since it has been observed that people tend to wait for others to finish taking a selfie before taking their turn. This lets selfie takers successfully finish their photo session while other people are still taking pictures of Sophia.

- **Gaze Angle**

  The gaze direction of detected people was taken into account in Photogenic Behavior, as a contributor to rank inheritance.

  - `posing` - the gaze angle penalty was set to be more significant, as people usually look in the direction of their picture-taking focus.
  - `selfie` - since Sophia usually cannot see the person she's taking a selfie with, the gaze angle penalty is neglected.

An example of an experiment involving gaze angle is shown in Figure 3.1, where two people were asked to stand in front of Sophia at similar distance and FOV angle, but only one of them was looking at her. The person looking at Sophia was given a higher rank than the other person. This experiment shows that the attention system is tuned so that when *Person 1* is turned away, Sophia prioritizes *Person 2* who is facing towards her even though their face is outside the FOV.

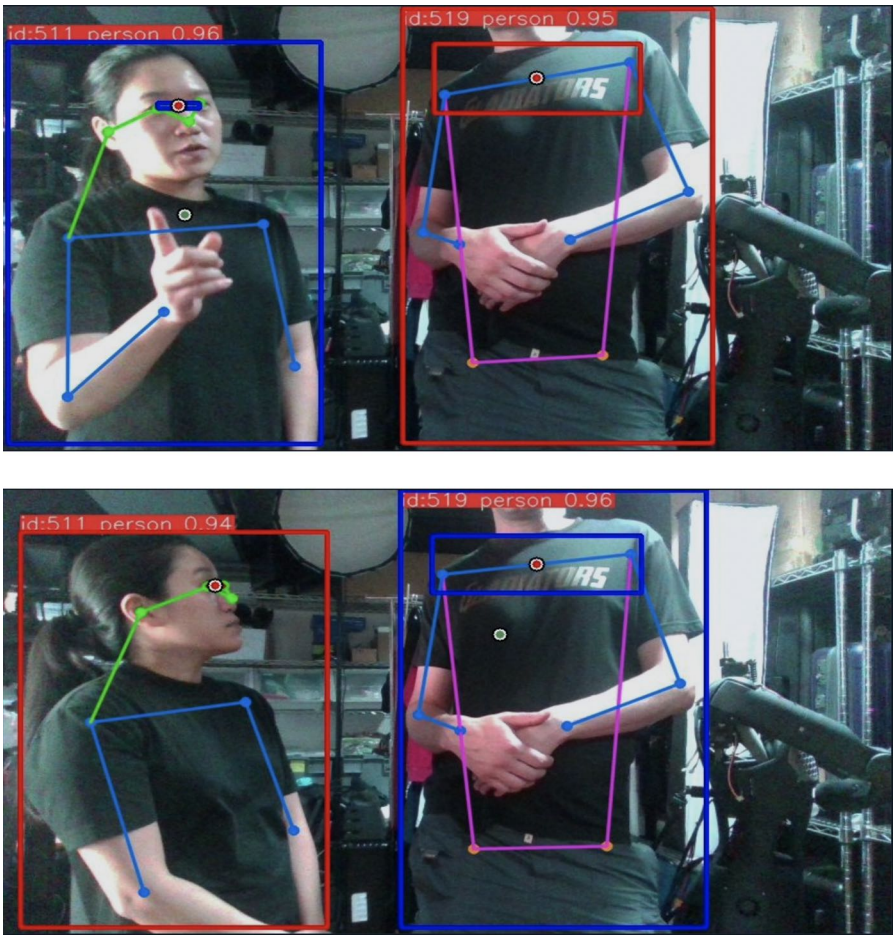


Figure 3.1: Testing the `gaze_angle_penalty` mechanism. From left to right: *Person 1* and *Person 2*. *(Source: Author and Hanson Robotics Limited)*

- **Action Recognition**

  Specific actions like taking a photo or waving were given higher ranks in the `posing` state. Waving has been set to have a higher `max_rank` and `confidence_multiplier` than taking a photo as it is a direct signifier of attention seeking, while taking a photo is a more passive action.

- **Sound Source Localization**

This feature was mainly used in the `posing` state, where Sophia could accurately match the sound source to a visible person. This was not used in the `selfie` state, as the sound source is usually very close, not visible to Sophia, and at a dramatically different angle than the PTD.

- **Phrase Detection**

  Key phrases related to taking photos were identified and used to trigger Photogenic Behavior.

- **Attention Inheritance**

  The influence of nearby people on the attention rank of PTDs was again considered mainly in the `posing` state. The `inherit_attention` feature was set to have a high `multiplier` value as a significant portion of the attention is inherited from the person holding the PTD.

  An example of an experiment involving attention inheritance is shown in Figure 3.2, where two people were both holding a phone and taking pictures of Sophia, while performing certain actions, moving around, and talking. The participants were observing if Sophia's attention system reacted appropriately to their actions while focusing on their phones.

Figure 2.18 shows the attention system settings for the `posing` state as an example.

3.2 Questionnaire and Feedback

The evaluation included a questionnaire focused on various aspects of Photogenic Behavior. Participants provided feedback on Sophia's ability to recognize picture-taking intentions, track PTDs, and appropriately enter or avoid Photogenic Behavior. Each question was rated on a scale from 1 to 5, with 1 being the lowest (*total disagreement*) and 5 being the highest score (*total agreement*).

The participants were asked to act out various scenarios of picture-taking, including taking selfies with Sophia and taking pictures of her, both in groups from 1 to 5 people and individually. After going through at least 5 different scenarios, the participants were asked to fill out the questionnaire, divided into three main sections with specific subquestions:

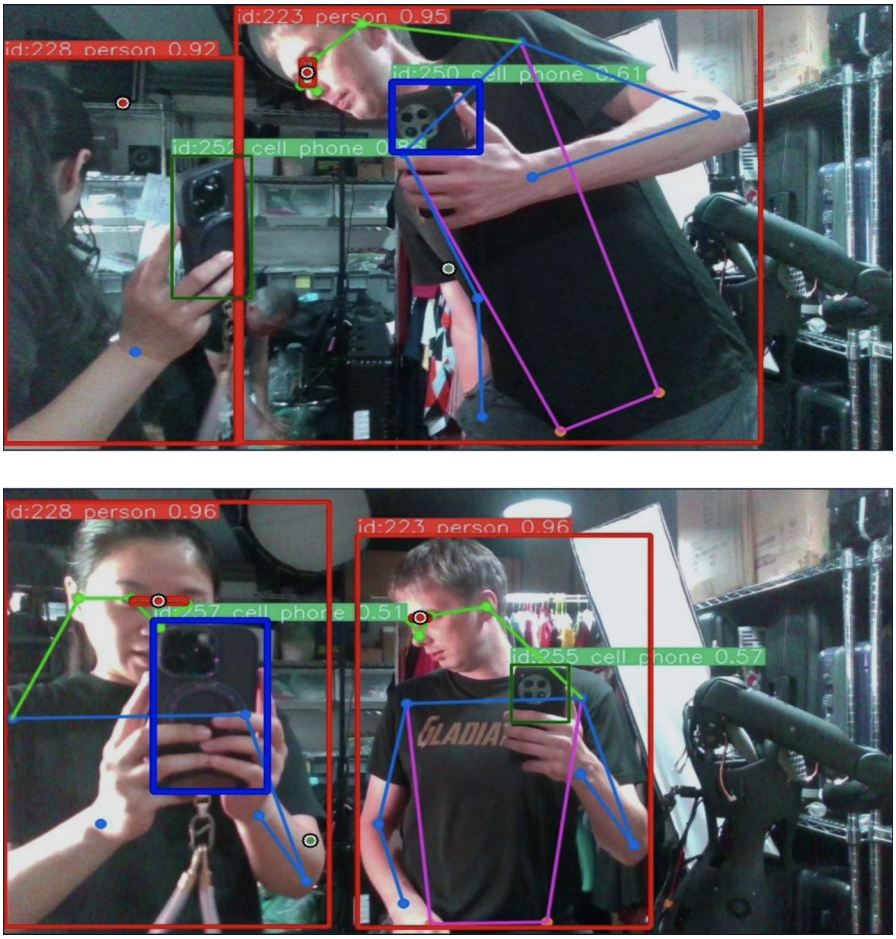


Figure 3.2: Testing the `attention_inheritance` mechanism. From left to right: *Person 1* and *Person 2. (Source: Author and Hanson Robotics Limited)*

- **Recognition of Picture-Taking Intention**

  These questions focused on Sophia's ability to enter an appropriate Photogenic Behavior state.

  - Recognized intention based on conversation cues. *(e.g., context of a conversation, or a direct command like "Let's take a picture.")*
  - Recognized intention based on nonverbal cues. *(e.g., holding a PTD, and waving at Sophia)*
  - Verbally indicated awareness of being photographed when appropriate. *(e.g., saying "Cheese" or "Let's take a picture" when a PTD is detected)*
  - Smiled and/or posed appropriately for a picture.

- **Tracking Picture-Taking Devices**

- Gazed at a PTD.
- Accurately tracked a moving PTD.
- Looked at the most appropriate PTD when multiple were present.
- Switched to another PTD appropriately, after giving attention to the first PTD for a reasonable time.

- **Recognition of Inappropriate Situations for Photogenic Behavior**
  - Didn't look at a PTD held by a person when engaged in a conversation with another person.
  - Didn't look at a PTD during a scripted performance/scenario that didn't involve Photogenic Behavior.

Table 3.1 summarizes the results of the questionnaire. The feedback was generally positive, indicating successful calibration and implementation of the Photogenic Behavior.

Question	**Mean**	**SD**	**Max**	**Min**	**POS** [%]	**NEG** [%]
Recognized intention (conversation cues)	4.27	0.68	5	3	87	0
Recognized intention (nonverbal cues)	3.40	0.71	4	2	53	13
Responded verbally	3.67	1.40	5	1	73	27
Smiled/posed appropriately	4.33	0.60	5	3	93	0
Gazed at a PTD	4.47	0.62	5	3	93	0
Accurately tracked a moving PTD	4.00	0.63	5	3	80	0
Looked at the most appropriate PTD	4.07	0.93	5	2	73	7
Switched to another PTD	3.53	1.26	5	1	67	20
Didn't look at PTD (engaged in conversation)	3.53	0.81	5	2	60	13
Didn't look at PTD (scripted performance)	4.33	0.70	5	3	87	0
Average	**3.96**	**0.83**	**4.9**	**2.3**	**77**	**8**

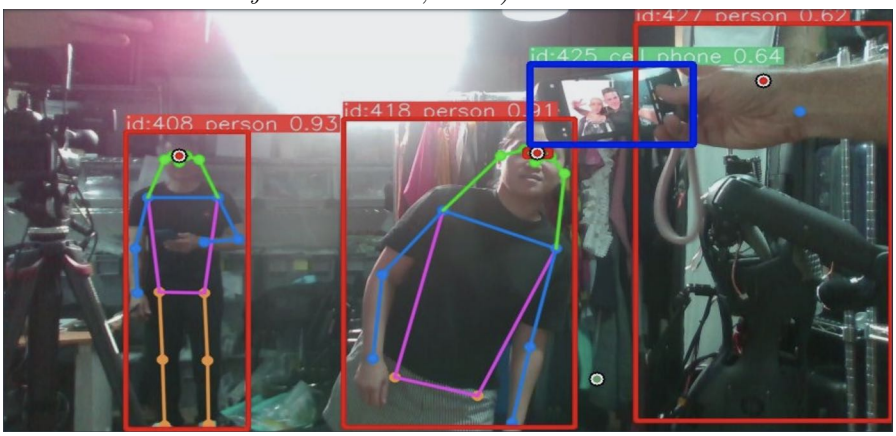
Table 3.1: Results of the questionnaire evaluating Sophia's Photogenic Behavior.

3.3 Results and Discussion

The iterative calibration of Sophia's attention system proved effective, resulting in an lifelike ability to recognize and respond to picture-taking scenarios. Each feature's importance varied across different scenarios, highlighting the need for a flexible and context-aware approach. The feedback from participants was overwhelmingly positive, indicating that the enhancements successfully achieved the desired objectives.

An crucial part of judging the quality of Photogenic Behavior are the resulting pictures taken by the participants. The participants were asked to take pictures of and with Sophia in various scenarios. The results were generally positive, with Sophia looking into the camera in most pictures, smiling appropriately, and occasionally posing for the camera when appropriate. An example of a picture taken by a participant is shown in Figure 3.3a. Sophia's point of view during the picture taking, as seen in the HRSDK WebUI, is shown in Figure 3.3b.

(a) : Dr. David Hanson taking a selfie with Sophia. *(Source: Provided by David Hanson, Ph.D.)*



(b) : Sophia's point of view during the selfie shoot. *(Source: Author and Hanson Robotics Limited)*

Figure 3.3: Sophia's creator, Dr. David Hanson, testing the new Photogenic Behavior module.

The largest standard deviation and percentage of negative responses was observed in the *Responded verbally* question, which was expected due to the optional nature of this aspect of Photogenic Behavior. The second largest standard deviation and percentage of negative responses was in the *Switched to another PTD* question, which was due to the

complexity of the scenario and the subjectivity of the participants' expectations. The participants' feedback was generally positive, with a mean score of 3.96 out of 5, indicating a high level of satisfaction with the current configuration of Sophia's Photogenic Behavior. This score can be further improved by refining the system based on larger dataset and more diverse scenarios. A proposal for future work is to conduct a similar evaluation with a larger group of participants, in a public setting where this behavior would be more commonly used.

Another approach would be to reconstruct picture-taking scenarios in a controlled environment first for Sophia, then for human participants in her place, and compare the resulting pictures taken. This would provide a more direct comparison of Photogenic Behavior between Sophia and humans, and it is more readily achievable, yet to achieve the same level of objectivity it would be extremely time-consuming since it would require a large number scenario recreations.

3.4 Comparisons

Compared with Sophia's initial state, the enhancements made in this work are prominent. For illustration: the main changes in the codebase were made to the GitHub repositories listed below. Over 5000 lines of code in total were modified in these repositories, with the majority of the changes being in the `r2_behavior` repository.

- `r2_perception` - this repository is the perception ROS package of Sophia [58].
- `r2_behavior` - this repository is a ROS package with the behavior and attention ROS nodes of Sophia [59].
- `hr_msgs` - this repository contains the ROS messages utilized by Hanson Robotics products [60].
- `hrsdk_configs` - this repository contains various configuration files used by the system [61].

The general changes made to the system can be summarized as follows, while the broader contributions are discussed in the next chapter:

- *Upgraded Perception Abilities*: Initially, Sophia could only perceive humans. The addition of state-of-the-art real-time detection models and new multimodal LLMs has significantly improved Sophia's ability to recognize and respond to picture-taking scenarios. The additions of abilities to detect and track any kind of object and recognize peoples' actions have made Sophia more context-aware and capable of responding appropriately in a wider range of scenarios.
- *Novel Attention System*: Previously, Sophia could only choose a person to gaze at randomly. The new attention system has been designed to be context-aware, adaptive, tunable, and capable of handling a wide range of scenarios and object classes in a human-like way.
- *Added Photogenic Behavior*: The new Photogenic Behavior states, `selfie` and `posing`, have been implemented to demonstrate the capabilities of the new attention system. The system has been calibrated and evaluated through a questionnaire and feedback from participants, demonstrating its effectiveness in recognizing and responding to picture-taking scenarios.
- *Suggested Hardware Enhancements*: The proposed hardware enhancements, including a wider FOV camera array and a more advanced microphone array, will further improve Sophia's perception capabilities and enable her to respond more naturally in a wider range of scenarios. The new system has been designed and tested to be compatible with these enhancements, ensuring a seamless transition.

Comparing this work to a GCS and its evaluation done by Aliasghari, Taheri, et al. [21] where they also focused on human-robot interaction (HRI) using a humanoid robot, similar iterative processes were employed to refine robot behaviors based on user feedback and a questionnaire was used to evaluate the system. That study, as well as another one by Zaraki, Mazzei, et al. [20] conducted on another Hanson Robotics robot 'FACE', used human test subjects and measured their gaze behavior using head-tracking and eye-tracking technology, while playing back a recording from the robot's point of view. The results were then compared to the robot's gaze behavior to evaluate the system's performance. This would be an interesting avenue for future testing on Sophia, as it would provide a more objective measure of the system's performance, however it would require additional hardware and setup to track the participants' gaze.

However, the contributions of my work improve upon these studies by focusing on a more complex and context-aware system that can handle a wider range of scenarios and interactions. The integration of advanced zero-shot real-time detection models, multimodal LLMs, and an easily adjustable GCS enables Sophia to respond more naturally and appropriately to picture-taking scenarios, and lays foundation for a universal GCS adaptable to a wide range of scenarios and robots.

3.5 Challenges and Limitations

While the calibration and evaluation process demonstrated the robustness of the new system, several challenges and limitations were encountered:

- **Computational Load**

  The real-time processing of multiple models and sensors occasionally led to delays, suggesting a need for further optimization or more powerful hardware.

  This can be addressed on the software side by optimizing the code, opting for smaller models, or using optimization techniques like model quantization or platform specific optimization frameworks like the Nvidia TensorRT [62].

  On the hardware side, this could be addressed by upgrading the processing unit to a more powerful one or offloading some of the processing to a separate unit. Adding more processing units to the system, might noticeably impact Sophia's battery life, so a balance between performance and power consumption would have to be struck.

- **Perception Limitations**

  Changes in lighting and background noise affected the performance of visual and auditory detection models respectively, indicating a need for further hardware improvements. Another current limitation is the narrow FOV of Sophia's main camera, which has been partially addressed in this work by proposing a temporary enhancement to Sophia's camera array until a better solution can be implemented.

  This can be improved by using more advanced sensors, with higher dynamic range and noise reduction capabilities. The main limitation here is the cost of such sensors, as well as the limited real estate on Sophia's body which is usually covered by clothes and only leaves a small area for sensors.

  For increasing FOV in a more streamlined way, a custom stereoscopic camera setup could be designed. A similar setup was used in the StereoPi project [63], which achieved reasonably accurate depth perception and a wide FOV using a pair of fisheye cameras.

- **User Variability**

  Different participants had varying expectations and interpretations of Sophia's behaviors, which sometimes led to inconsistent feedback. This highlights the need for a more diverse and larger sample size in future evaluations.

Chapter 4

Conclusion and Future Work

4.1 Contributions

- **Integration with Current Hanson Robotics Platform**

  The enhancements have been integrated with Sophia's existing software and hardware architecture, ensuring compatibility with the current Hanson Robotics platform and their SDK software (HRSDK). This integration ensures a seamless transition to the new system and enables the enhancements to be deployed on any existing Sophia platform without significant modifications.

  The whole system is built on top of the Robot Operating System (ROS), which allows for easy integration with other robotic platforms and provides a standardized environment for developing and testing the system.

  Over 5000 lines of code in total were added or modified in various Hanson Robotics GitHub repositories [58, 59, 60, 61]. Archived snapshots of these repositories can be found in Appendix B.

- **Modularity and Scalability**

  The new system is designed to be modular and scalable, allowing for the addition of new features and behaviors in the future, and potential adaptation to other humanoid robots.

  Likewise, it is designed to work on Sophia's current hardware, even though hardware modifications are proposed to support the enhanced sensory requirements. This also

ensures that the cost of implementing and testing the new system is minimized, as it can be deployed on existing hardware with minimal changes.

- **Enhanced Perception Capabilities**

  Implementing advanced real-time detection models (YOLOv8-Pose, YOLOv9) [33] [64], allows Sophia to recognize and track not only people but also other objects in her environment, including but not limited to cameras, video cameras, cellphones, and other notable items. These can be either from the standard COCO dataset labels [4], custom-trained for specific events, or even state-of-the-art models for real-time zero-shot detection [65] specifically YOLO-World created by Cheng, Song, et al. [66], enabling Sophia to detect any object classes not present in the training dataset.

  Sophia is now able to recognize human actions [42] such as taking a photo, waving, or pointing, based on the 60 actions specified in the NTU RGB+D dataset [67]. This enables her to respond appropriately to these actions, focusing her attention appropriately.

  With the addition of a directional microphone array [1], Sophia is now able to localize and separate sound sources, which can be used to match speech to the person speaking, or to detect and respond to specific sounds, such as applause or laughter.

  Sophia's field of view (FOV) will be expanded from the current 69° to 160° [68] through the addition of a wide-angle fisheye camera [2] in tandem with her current Intel RealSense RGB-D camera [27], serving as Sophia's peripheral vision. This significantly enhances her situational awareness and enables her to detect and respond to visual stimuli outside her direct line of sight.

- **Autonomous Gaze Control System**

  A highly customizable GCS has been developed and tuned, enabling Sophia to autonomously control her gaze and attention based on the perceived stimuli. The GCS is integrated with Sophia's behavior system, which has also been enhanced, and which regulates the GCS by modifying its configuration based on the current context and the desired behavior.

  The GCS extracts a number of features of each perceived stimulus, ranks them based on the current context and generates a gaze plan that specifies where Sophia should look and how she should behave.

  At the same time, the GCS is designed to be transparent and interpretable, allowing operators and developers to understand and modify its behavior easily. This transparency is essential for ensuring that Sophia's interactions are predictable, controllable, and adaptable to new scenarios.

- **Integration with Sophia's AI agents**

Sophia uses a state-of-the-art multi-agent system (MAS) for her conversation and reasoning capabilities, utilizing latest large language models (LLMs), retrieval-augmented generation (RAG), and advanced prompting techniques, controlled by an agent supervisor. This network has been integrated with the new system, enabling it to provide context-aware information to the GCS and behavior system. This helps Sophia to make more informed decisions about her gaze and attention, based on the current conversation and the context of the interaction.

Proposed additions to the AI agent network include a locally running multimodal LLM, such as LLaVA created by Liu, Li, et al. [53] [54], which can provide low latency descriptions of the scene and the objects and people in it.

- **Real-world Evaluation and User Feedback**

  A user poll was conducted to gather feedback on the current system and to identify areas for improvement. The results of the poll were used to guide the development of the new system and to prioritize the enhancements that would have the greatest impact on Sophia's interactions.

  The new system has been tested and evaluated on real-world scenarios. User feedback was collected and analyzed to assess the effectiveness of the new system and to identify areas for improvement. The feedback was used to refine the system and to make it more responsive and engaging in social settings, namely in picture-taking scenarios.

4.2 Future Development

A significant portion of the future work for has already been outlined and discussed throughout this thesis. The following are key areas of focus and anticipated developments:

- **Hardware Enhancements**
  - Several hardware improvements have been proposed and are currently in the prototype stage, like the directional microphone array, fisheye camera, and movable waist mechanism.
  - The near-term goal is to complete testing and make these enhancements production-ready, ensuring they integrate seamlessly with Sophia's existing systems.
- **Software Improvements**

- **Optimization and Refinement**
  - *Performance Optimization*: While the code has been designed to be robust and stable, there is always room for speed optimization. Improving the efficiency of the code will be a priority to ensure smooth and responsive interactions, especially in real-time scenarios.
  - *Configuration Tuning*: Continuous tuning and refinement of the configuration files will be necessary to adapt to new use cases and environments. This iterative process will help in maintaining the system's flexibility and responsiveness. *Testing and Feedback*: To ensure quality of the system, extensive testing with a larger pool of participants will be crucial. Sophia's frequent appearances at public events will provide ample opportunities for real-world testing and feedback collection.
- **New Modes and Machine Learning**
  - *Utilizing the Modular GCS*: The GCS has been designed to be modular and adaptable. Exploring new ways to utilize this system, such as enabling the MAS to decide which specific object or person to track based on conversation cues, will be a key area of development.
  - *Machine Learning Integration*: Implementing a machine learning model to tune the attention system using supervised learning techniques could provide significant improvements. By training the system on a large dataset of human interactions, it could learn to prioritize attention cues more effectively and adapt to new scenarios with minimal manual configuration.

- **Practical Applications**
  - **Human-Robot Interaction (HRI) Applications**
    - *Entertainment and Customer Service*: The Sophia platform can be deployed in environments such as entertainment venues, retail stores, and customer service centers. Her ability to engage with customers, provide information, and entertain makes her a valuable asset in these settings.
    - *Healthcare and Elderly Care*: In healthcare settings, particularly in elderly care, Sophia can provide social interaction, remind patients of their medication schedules, and assist in monitoring their well-being, supporting human staff who are often stretched thin.

The enhancements made in this thesis, along with the proposed future developments, not only contribute to the improvement of Sophia but also to the broader field of social robotics, playing part in the integration of robots into human society.

Appendix A

Bibliography

[1] "iFLYTEK Far-Field Microphone Array Module ROS Six-Microphone Voice Boa — hiwonder.com." https://www.hiwonder.com/products/far-field-microphone-array-module. [Accessed 12-05-2024].

[2] "See3cam-cu81-ar0821 4k hdr usb camera." https://www.e-consystems.com/usb-cameras/ar0821-8mp-4k-hdr-camera.asp.

[3] A. Mustofa, "Yolov8 pose estimation and pose keypoint classification using neural net pytorch." https://alimustoofaa.medium.com/yolov8-pose-estimation-and-pose-keypoint-classification-using-neural-net-pytorch-98469b924525, June 2023.

[4] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft COCO: Common Objects in Context." https://arxiv.org/abs/1405.0312, 2015.

[5] C. Ball, *Converge: a futurist's insights into the potential of our world as technology and humanity collide.* Major Street Publishing, 2022.

[6] "Does AI creep you out? You're experiencing the 'uncanny valley' — nationalgeographic.com." https://www.nationalgeographic.com/science/article/ai-uncanny-valley. [Accessed 13-05-2024].

[7] Z. Stone, "Everything You Need To Know About Sophia, The World's First Robot Citizen." https://www.forbes.com/sites/zarastone/2017/11/07/everything-you-need-to-know-about-sophia-the-worlds-first-robot-citizen/, 11 2017.

[8] C. Monteiro, "UNDP in Asia and the Pacific Appoints World's First Non-Human Innovation Champion." https://web.archive.org/

web/20180709173848/http://www.asia-pacific.undp.org/content/rbap/en/home/presscenter/pressreleases/2017/11/22/rbfsingapore.html. [Accessed 08-05-2024].

[9] J. Retto, "Sophia, first citizen robot of the world." https://www.researchgate.net/publication/321319964_SOPHIA_FIRST_CITIZEN_ROBOT_OF_THE_WORLD, 11 2017.

[10] J. Fernandes, "Robot citizenship and gender (in)equality: the case of sophia the robot in saudi arabia," *JANUS NET e-journal of International Relation*, vol. 01, 02 2022.

[11] R. Moberg and A. Khan, "Humanoid robot acceptance: A concise review of literature," in *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1223–1228, 2022.

[12] S. Mukherjee, M. M. Baral, S. K. Pal, V. Chittipaka, R. Roy, and K. Alam, "Humanoid robot in healthcare: A systematic review and future research directions," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, vol. 1, pp. 822–826, 2022.

[13] F. Kirstein and R. V. Risager, "Social robots in educational institutions they came to stay: Introducing, evaluating, and securing social robots in daily education," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 453–454, March 2016.

[14] T. Iwamoto, J. Baba, J. Nakanishi, K. Hyodo, Y. Yoshikawa, and H. Ishiguro, "Playful recommendation: Sales promotion that robots stimulate pleasant feelings instead of product explanation," *IEEE Robotics and Automation Letters*, vol. 7, pp. 11815–11822, Oct 2022.

[15] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "A communication robot in a shopping mall," *IEEE Transactions on Robotics*, vol. 26, pp. 897–913, Oct 2010.

[16] A. Hong, N. Lunscher, T. Hu, Y. Tsuboi, X. Zhang, S. Franco dos Reis Alves, G. Nejat, and B. Benhabib, "A multimodal emotional human–robot interaction architecture for social robots engaged in bidirectional communication," *IEEE Transactions on Cybernetics*, vol. 51, pp. 5954–5968, Dec 2021.

[17] B. Goertzel, J. Mossbridge, E. Monroe, D. Hanson, and G. Yu, "Humanoid robots as agents of human consciousness expansion." https://arxiv.org/abs/1709.07791, 2017.

[18] V. Somashekarappa, "Implementation of gaze estimation in dialogue to human-robot interaction," pp. 1–3, 10 2022.

[19] S.-S. Yun, "A gaze control of socially interactive robots in multiple-person interaction," *Robotica*, vol. 35, p. 2122–2138, Nov. 2017.

[20] A. Zaraki, D. Mazzei, M. Giuliani, and D. De Rossi, "Designing and evaluating a social gaze-control system for a humanoid robot," *IEEE Transactions on Human-Machine Systems*, vol. 44, p. 157–168, Apr. 2014.

[21] P. Aliasghari, A. Taheri, A. Meghdari, and E. Maghsoodi, "Implementing a gaze control system on a social robot in multi-person interactions," *SN Applied Sciences*, vol. 2, p. 1135, May 2020.

[22] "Meet sophia, the robot that looks almost human." `https://www.nationalgeographic.com/photography/article/sophia-robot-artificial-intelligence-science`, May 2018.

[23] J. Kasbe and C. Moselle, "Sophia." `https://www.imdb.com/title/tt19757646/`, 2022. Film, Documentary.

[24] S. Yifan, X. Mo, V. Krisciunas, D. Hanson, and B. E. Shi, "Intention estimation via gaze for robot guidance in hierarchical tasks," in *Proceedings of The 1st Gaze Meets ML workshop* (I. Lourentzou, J. Wu, S. Kashyap, A. Karargyris, L. A. Celi, B. Kawas, and S. Talathi, eds.), vol. 210 of *Proceedings of Machine Learning Research*, pp. 140–164, PMLR, 03 Dec 2023.

[25] D. F. Hanson, "Human emulation robot system." `https://patents.google.com/patent/US7113848B2/en`, Sept. 2006. US Patent 7,113,848.

[26] A. Imran, D. Hanson, G. Morales, and V. Krisciunas, "Open arms; open-source arms, hands and control," in *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pp. 1426–1431, Nov 2022.

[27] "Depth Camera D435i — intelrealsense.com." `https://www.intelrealsense.com/depth-camera-d435i/`. [Accessed 13-05-2024].

[28] "Intel NUC 13 Pro: Small Outside, Powerful Inside — intel.com." `https://www.intel.com/content/www/us/en/newsroom/news/intel-nuc13-pro-small-outside-powerful-inside.html`. [Accessed 12-05-2024].

[29] "NVIDIA Jetson Orin — nvidia.com." `https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/`. [Accessed 12-05-2024].

[30] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sept. 2016.

[31] Pallawi, "Which human pose estimation model should you pick to realise your ideas for a video analytics. . . ." `https://pallawi-ds.medium.com/`

which-human-pose-estimation-model-should-you-pick-to-realise-your-ideas-for-a-video-analytics-6ca754cc1f4e, Dec. 2023.

[32] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO." https://github.com/ultralytics/ultralytics, Jan. 2023.

[33] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, "A review on yolov8 and its advancements," in *Data Intelligence and Cognitive Informatics* (I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, eds.), pp. 529–545, Springer Nature Singapore, 2024.

[34] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking." https://arxiv.org/abs/2206.14651, 2022.

[35] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box." https://arxiv.org/abs/2110.06864, 2022.

[36] M. R. Munawar, "Ultralytics yolov8 object trackers (botsort vs bytetrack) comparison." https://muhammadrizwanmunawar.medium.com/ultralytics-yolov8-object-trackers-botsort-vs-bytetrack-comparison-d32d5c82ebf3, Apr. 2024.

[37] W. Fuhl, D. Weber, and S. Eivazi, "Groupgazer: A tool to compute the gaze per participant in groups with integrated calibration to map the gaze online to a screen or beamer projection." https://arxiv.org/abs/2201.07692, 2023.

[38] A. Kottwani and A. Kumar, "Eye gaze estimation model analysis." http://arxiv.org/abs/2207.14373, July 2022. arXiv:2207.14373 [cs].

[39] A. Maltsev, "Action recognition in the wild." https://medium.com/@zlodeibaal/action-recognition-in-the-wild-9eb7f12b4d12, Apr. 2023.

[40] MMAction2 Contributors, "OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark." https://github.com/open-mmlab/mmaction2, July 2020.

[41] A. Paszke and S. G. et al., "Pytorch: An imperative style, high-performance deep learning library." https://arxiv.org/abs/1912.01703, 2019.

[42] H. Duan, J. Wang, K. Chen, and D. Lin, "Pyskl: Towards good practices for skeleton action recognition." https://arxiv.org/abs/2205.09443, 2022.

[43] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis." https://arxiv.org/abs/1604.02808, 2016.

[44] "Xf Mic Asr Offline · GitLab — git.bwbot.org." https://git.bwbot.org/publish/xf_mic_asr_offline. [Accessed 15-05-2024].

[45] J. E. Goldring, M. C. Dorris, B. D. Corneil, P. A. Ballantyne, and D. P. Munoz, "Combined eye-head gaze shifts to visual and auditory targets in humans," *Experimental Brain Research*, vol. 111, p. 68–78, Sept. 1996.

[46] T. J. Buschman and E. K. Miller, "Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices," *Science (New York, N.Y.)*, vol. 315, p. 1860–1862, Mar. 2007.

[47] L. O. M. Rothkegel, H. A. Trukenbrod, H. H. Schütt, F. A. Wichmann, and R. Engbert, "Temporal evolution of the central fixation bias in scene viewing," *Journal of Vision*, vol. 17, p. 3, Nov. 2017.

[48] R. F. Thompson and W. A. Spencer, "Habituation: a model phenomenon for the study of neuronal substrates of behavior," *Psychological Review*, vol. 73, p. 16–43, Jan. 1966.

[49] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.

[50] "meta-llama (Meta Llama) — huggingface.co." https://huggingface.co/meta-llama. [Accessed 16-05-2024].

[51] O. et al., "Gpt-4 technical report." https://arxiv.org/abs/2303.08774, Mar. 2024. arXiv:2303.08774 [cs].

[52] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge." https://llava-vl.github.io/blog/2024-01-30-llava-next/, January 2024.

[53] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning." http://arxiv.org/abs/2304.08485, Dec. 2023. arXiv:2304.08485 [cs].

[54] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning." http://arxiv.org/abs/2310.03744, Oct. 2023. arXiv:2310.03744 [cs].

[55] "GitHub - ollama/ollama: Get up and running with Llama 3, Mistral, Gemma, and other large language models. — github.com." https://github.com/ollama/ollama. [Accessed 18-05-2024].

[56] "GitHub - google-research/big_vision: Official codebase used to develop Vision Transformer, SigLIP, MLP-Mixer, LiT and more. — github.com." https://github.com/google-research/big_vision. [Accessed 18-05-2024].

[57] D. Hanson, A. Imran, A. Vellanki, and S. Kanagaraj, "A neuro-symbolic humanlike arm controller for sophia the robot." https://www.researchgate.net/publication/344910756_A_Neuro-

Symbolic_Humanlike_Arm_Controller_for_Sophia_the_Robot, 10 2020.

[58] V. Krisciunas, W. Huang, and J. Sura, "GitHub - r2_perception: Unified perception system for Hanson Robotics — github.com." https://github.com/hansonrobotics/r2_perception. [Accessed 23-05-2024].

[59] V. Krisciunas, M. H. Leung, W. Huang, and J. Sura, "GitHub - r2_behavior: Behavior system for nonverbal communication, eye contact, listening/speaking, etc. — github.com." https://github.com/hansonrobotics/r2_behavior. [Accessed 23-05-2024].

[60] V. Krisciunas, M. H. Leung, W. Huang, and J. Sura, "GitHub - hr_msgs: ROS message definition for Hanson Robotics — github.com." https://github.com/hansonrobotics/hr_msgs. [Accessed 23-05-2024].

[61] H. Robotics, "GitHub - hrsdk_configs: Configuration for Hanson Robotics — github.com." https://github.com/hansonrobotics/hrsdk_configs. [Accessed 23-05-2024].

[62] Nvidia, "GitHub - NVIDIA/TensorRT: NVIDIA® TensorRT™ is an SDK for high-performance deep learning inference on NVIDIA GPUs. This repository contains the open source components of TensorRT. — github.com." https://github.com/NVIDIA/TensorRT. [Accessed 23-05-2024].

[63] "GitHub - realizator/stereopi-fisheye-robot: Python stereoscopic robot vision." https://github.com/realizator/stereopi-fisheye-robot. [Accessed 09-05-2024].

[64] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information." https://arxiv.org/abs/2402.13616, 2024.

[65] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection." https://arxiv.org/abs/1804.04340, 2018.

[66] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," *arXiv preprint arXiv:2401.17270*, 2024.

[67] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding." https://arxiv.org/abs/1905.04757, Oct. 2020.

[68] "Intel RealSense D400 Series Product Family Datasheet — dev.intelrealsense.com." https://dev.intelrealsense.com/docs/intel-realsense-d400-series-product-family-datasheet. [Accessed 12-05-2024].

Appendix B

Code Archive

This appendix contains the source code for the software extension developed in this thesis. The code comes appended in a single ZIP file with three folders representing the respective GitHub repositories: `r2_perception`, `r2_behavior`, and `hr_msgs`. Each folder contains the complete ROS package with the source code and launch files. The code is provided under a license agreement between Czech Technical University in Prague and Hanson Robotics Limited.