Gaze direction in the context of social human-robot interaction

Benoît Massé

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel : 25 Mai 2016

Présentée par

Benoit Massé

Thèse dirigée par **Radu Horaud**

préparée au sein **INRIA Grenoble Rhône-Alpes**
et de **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Étude de la direction du regard dans le cadre d'interactions sociales incluant un robot

Gaze Direction in the context of Social Human-Robot Interaction

Thèse soutenue publiquement le **29 Octobre 2018**,
devant le jury composé de :

Pr. Adrien Bartoli
Université Clermont Auvergne, Rapporteur
Dr. Mathieu Salzmann
École Polytechnique Fédérale de Lausanne, Rapporteur
Dr. Sileye Ba
Dailymotion Paris, Examinateur
Dr. Hayley Hung
Technical University of Delft, Examinatrice
Dr. Radu Horaud
INRIA Grenoble Rhône-Alpes, Directeur de thèse
Dr. Edmond Boyer
INRIA Grenoble Rhône-Alpes, Président

Abstract

Robots are expected to be more and more present in our everyday environment. They are likely not only to share physical spaces with humans but also to interact with them. In this context, robots are expected to understand both verbal and non-verbal cues, some of which being ambiguous, routinely used in natural human-to-human interactions. In particular, *gaze direction* (where are people looking at?), and *visual focus of attention* or *VFOA* (to whom or to what are people looking at?), are very valuable sources of information to understand the social behavior of each person, as well as the inter-person interaction dynamics. To estimate the VFOAs, the robot must solve multiple tasks. (a) Find and keep people in its camera field of view. The robot needs a suitable gaze control strategy, *i.e.* a strategy that uses sensory information to move its camera. (b) Estimate people gaze directions. The participants are expected to frequently look either at each other or at an object of interest; therefore their eyes are not always visible. Gaze estimation based on eye image patch is unreliable. However, the correlation between eye gaze and head movements can be exploited. (c) Locate the objects of interest. When the locations of objects of interest are unknown and outside the camera field of view, the presence of such objects can only be detected by following the gaze of participants. (d) Combine these data to derive the VFOAs.

In this thesis, we address the problem of simultaneously estimating the visual focus of attention of multiple people involved in a social interaction, from the point-of-view of an active humanoid robot. Along the way, we address the problem of robot gaze control, and the detection of out-of-view objects from gaze following. The proposed contributions are data-driven and are detailed as follows.First, we suppose that the locations of objects of interest are known. In this context, we model the gaze behavior with a Bayesian network, using findings from psychophysics. More precisely, we introduce a temporal model that describes the dependency between head poses, object locations, eye-gaze directions, and VFOAs. The proposed formulation is based on a switching linear dynamical system. It leads to a tractable learning procedure and to an efficient algorithm that simultaneously tracks gaze and VFOA. Second, we propose a model able to locate objects by combining people's gaze directions over time. The sequence of head poses is encoded into a heat-map representation adopting a top-view perspective. We propose several encoder/decoder convolutional neural networks that predict object locations and compare them with heuristics and simpler learning approaches. Third, We propose a novel reinforcement learning method for robotic gaze control. The model is based on a recurrent neural network architecture to learn a value function. The robot autonomously learns a strategy for moving its head (and camera) using audio and visual observations. It is able to focus on groups of people in a changing environment. Finally, all contributions have been tested on publicly available datasets. Moreover, two methods that simulate synthetic scenarios are proposed for data augmentation, and are used for training and test.

Résumé

Les robots sont de plus en plus présents dans l'environnement quotidien. Il ne suffit plus de partager l'espace avec des humains, mais aussi d'interagir avec eux. Dans ce cadre, il est attendu du robot qu'il comprenne un certain nombre de signaux ambigus, verbaux et visuels, couramment utilisés pour communiquer entre humains. En particulier, la *direction du regard* (où les gens regardent-ils?) et la *cible d'attention visuelle* (qui ou quoi les gens regardent-ils?) contiennent beaucoup d'informations sur le comportement social individuel ainsi que sur la dynamique de groupe à l'oeuvre. Afin d'estimer la (ou les) cible d'attention visuelle, désignée par l'acronyme anglais *VFOA* pour *visual focus of attention*, le robot doit résoudre plusieurs tâches. (a) Trouver les gens et les garder dans le champ de vision. Le robot a besoin d'une stratégie appropriée de pilotage du regard pour orienter la caméra en fonction de ses données sensorielles. (b) Estimer la direction du regard de chacun. Les gens sont libres d'orienter la tête à leur convenance, ainsi les yeux ne sont pas toujours clairement visibles. Il n'est pas fiable de compter sur des images des yeux pour deviner l'orientation du regard. Toutefois, les mouvements de la tête et des yeux sont souvent liés, et cette corrélation peut être utilisée. (c) Repérer les objets d'intérêt. Un objet d'intérêt peut être en dehors du champ de vision de la caméra. Détecter la présence d'un tel objet peut seulement se faire en suivant les regards. (d) Combiner ces informations pour estimer les VFOAs.

Dans cette thèse, nous proposons une méthode pour estimer simultanément la cible d'attention visuelle (VFOA) de plusieurs personnes engagées dans un processus d'interaction sociale, depuis le point de vue d'un robot humanoïde. De plus, deux problèmes rencontrés en chemin ont attiré notre attention: piloter le regard du robot, et suivre le regard des gens pour détecter les positions des objets en dehors du champ de vision. Les différentes contributions, décrites en détail ci-après, reposent sur l'apprentissage automatique à partir de données. Premièrement, nous supposons connues les positions des objets d'intérêt. Dans ce cadre, nous modélisons la dynamique du regard avec un réseau Bayésien, en s'inspirant d'observations psychophysiques. Plus précisément, nous introduisons un modèle temporel qui décrit, dans un groupe de plusieurs personnes, les dépendances entre les têtes, les objets, les regards et les VFOAs. Ce modèle, basé sur un système Markovien à dynamiques multiples, permet d'obtenir une méthode d'apprentissage des paramètres ainsi qu'un algorithme efficace pour estimer, en continu, la direction du regard et le VFOA. Deuxièmement, nous proposons d'estimer la position des objets d'intérêt en combinant les regards au cours du temps. La succession des mouvements de tête de chacun est encodée sous forme de carte de chaleur vue du dessus. Nous avons élaboré et entraîné plusieurs réseaux de convolution de type encodeur/décodeur pour prédire les positions qui contiennent vraisemblablement des objets d'intérêt. D'autres méthodes plus simples sont présentées pour comparaison. Troisièmement, nous présentons une méthode d'apprentissage par renforcement pour piloter le regard du robot. Un réseau de neurones récurrents est entraîné pour prédire la valeur d'action. Le robot utilise ses observations audio et visuelles pour apprendre de manière autonome une stratégie efficace pour orienter sa tête. Cela lui permet de cibler des groupes de personnes dans un environnement évolutif. Enfin, toutes les contributions sont validées sur des jeux de données disponibles publiquement. De plus, deux méthodes de simulation de scénarios synthétiques ont été développées afin d'enrichir les jeux de données. Les scénarios générés peuvent être utilisés pour l'entraînement ou la validation.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

CHAPTER 1

INTRODUCTION

1.1 GENERAL CONTEXT

In recent years, there has been a growing interest in human-robot interaction (HRI), a research field dedicated to designing, evaluating and understanding robotic systems able to communicate with people. Such a robot can be used in many situations. In commercial places, the robot can serve as a waiter, a receptionist or a vendor that looks after clients' requests. There may also be a need for a companion robot that assists patients or elder people in their everyday life, or helps socially impaired people – like autistic children – experiment with social interaction. The evaluation of how good the robot is depends on objective criteria, *e.g.* the time required to address the client's request, and subjective criteria, like how natural did the interaction feel. The quality of human-robot social interactions depends on the robot taking actions that are appropriate to the situation. Understanding the environment, including implicit social and cultural cues, is the basis of an efficient decision making. In particular, mimicking human behavior is important for the robot so that people merely have to apply prior knowledge about human-to-human interactions. These challenges are addressed under a field named *social robotics*. Interestingly, we often imagine a social robot to be humanoid by similarity with actual human interaction, but this is not a hard constraint. For instance, a home automation system connected to a social artificial intelligence would be interesting to study. However, this would introduce an asymmetry between the robot and the people, and we would rather avoid to insert unnecessary bias in the interactions. From now on, unless specified otherwise, the term robot is used for *humanoid robot*.

In interpersonal communication, participants exchange messages using verbal and non-verbal signals, depending on the context. For a sighted person, a significant part of the external information processed by the brain comes from the visual system. *Gaze direction*, defined as the line between the eyes of the observer and the region he/she is focusing on, is a very informative visual cue. The interest of studying the gaze direction in social robotics is threefold. First, knowing the region of the space in which someone is interested helps understanding and predicting his/her actions, and more generally

gives strong insight on the context. For instance, a typical behavior is to look at the current speaker in a group or at the object currently being discussed. Second, switching gaze direction modifies the visual field. This is a very natural strategy to account for a lack of information. Third, human beings – even when still a toddler – are very efficient in estimating other people's gaze direction. This skill is used to communicate *e.g.* in joint attention mechanisms. For all these reasons, gaze direction is very useful for a social robot, either to better comprehend the environment, or to optimize its own decisions about where to look at.

In practice, for a robot, understanding the environment consists in being able to analyze data from its sensors and extract relevant information. Sensors typically include one or several of the following: RGB camera(s), microphones, depth sensor, self-motion sensor and/or force sensor. Analyzing multimodal data, and especially visual data, is a very complicated task to perform manually. The relationship between a set of pixels and a specific object is impossible for a human to handcraft. Indeed, the same thing can be represented by a prohibitively large number of images, sometimes very different from each other, depending on *e.g.* orientation, lighting, or background. For this reason, machine learning has been used for a long time to address this limitation. Machine learning is a research field whose goal is to extract patterns (*learn*) from data. It is used to solve different categories of problems.

- *Supervised learning* is when we have observations along with their associated expected decisions, and we want to predict the correct decision associated with new observations. In social robotics, it can be used for many tasks, *e.g.* predict whether there is a human in the field of view, which action he/she is doing or which direction he/she is facing. A training set of annotated examples is required for each of these problems. When the training labels are incomplete, noisy or partially missing, this is referred to as *semi-supervised learning*.
- *Unsupervised learning*, on the other hand, consists in modeling the underlying structure of the data without labels. A classical usage is *clustering*, *i.e.* grouping individuals given features. Robots sometimes need it in anomaly detection. It can also be used in the first stages of semi-supervised learning to complete or correct unreliable labels.
- *Reinforcement learning* is a particular weakly supervised learning method, popular in robotics. The true label is unknown but the decision can be evaluated a posteriori with a reward; the task is learned with trial-and-error. This is particularly relevant when evaluating the result of a sequence of actions. There are too many possible decision for the human supervisor to know the optimal one. This is useful *e.g.* for playing chess, or for some control problems.

Machine learning problems fall into different categories and can be tackled with various mathematical tools. The field has deeply and quickly evolved for the last few years with the impressive achievements of deep neural networks. Still, probabilistic formulations remain popular either alone or in conjunction with deep learning.

In this thesis, we have been interested in providing tools for a humanoid robot to progress toward a natural social interaction with humans by addressing several complementary challenges. In particular, we focus on determining which regions and which targets are interesting in a natural social scene along time, as well as the direction the robot should be looking at. The designed methods rely on mechanisms associated with gaze direction and operate in a data-driven framework using various machine learning techniques. Of course, understanding what is happening in social interactions is broader in scope than social HRI. It has applications in various domains such as video-surveillance, advertising or automatic reporting. A discussion about human and societal aspects of these topics is available in section 5.3.

1.2 VOCABULARY

There is an abundant literature studying gaze, both from the psychophysical and computer vision communities. However, even within a community, researchers often associate different meanings to the same word. We propose in this section to explicitly define the terms used throughout this thesis and their actual meanings.

We present first some geometrical terms, taking place in a fixed, global coordinate frame. **Position** and **Location** are generally synonyms and refer to the coordinates of a point in a predefined system. When speaking about a non-punctual object, they refer to the coordinates of its center of mass. In this work, we make a slight distinction between them. **Location** is used to speak about 3D coordinates while **Position** describes the 2D projection *e.g.* into an image or a top-view map. Inline clarifications (like 2D position) may be found when the context is ambiguous. The term **Orientation** refers to pan, tilt and roll angle of a rigid body. **Direction** represents a 3D unit vector and has two degrees of freedom. In general, direction and orientation are incompatible variables. However, in most cases detailed below, we do not need the roll angle; then the orientation is homogeneous to the direction spanned by two points of the rigid body's roll axis. As an example, see the relationship between head orientation and gaze direction explained below.

As seen before, **Gaze direction** is defined as the direction spanned by the line from the eyes of the observer to the region of space he/she is focusing on. It is sometimes called **Eye-gaze** to emphasize the role of the eyes. The object or person that lies in this region is the **Visual Focus of Attention** (**VFOA**) of the observer, also called the object of interest. **Head orientation** are the pan and tilt angle of the head (we drop the roll angle) with respect to a default orientation in the global 3D coordinate system. It is generally equated to the direction from the center of the head through the nose. An illustration of these concepts is available in Fig. 2.1 using notations from section 2.3.1. Then, gaze direction and head orientation can be expressed in the same space. The difference between them is entirely due to the orientation of the eyes w.r.t. the head. This difference is called **eyeball orientation**. Additionally, the term **Head pose** is often found in the literature and is short for joint head location and orientation.

Alternatively, a few other concepts are related to gaze. First, we use the term **Gaze**

following when we are interested in the position or location of the region being stared. This corresponds to the location of the VFOA. A particular case of gaze following is **on-screen gaze following**, when predicting the on-screen position of a person's focus. For instance, a smart-phone application could adapt to the user's gaze, knowing which pixels of the screen are being looked at. On a different note, **Gaze control** is a field of research from the neuroscience literature that studies all the mechanisms involved when a human or an animal switches gaze. Applied to robotics, **Gaze control** refers to the robot strategies for moving its cameras, and depends on the task.

Finally, the term **Full body pose** refers to the location and orientation of all limbs of a person. In practice, given constraints on the human body, it is considered sufficient to have the locations of the joints.

1.3 OVERVIEW

Hans Moravec wrote in the 1980's "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility" [82]. For more than 50 years, computer vision researchers have been working on extracting information from images and videos, yet machines still have not achieved human performance on many problems, for instance recognizing an object that has only been seen once. The most successful results have been achieved when combining computer vision with machine learning. Indeed, it is much easier to train a machine to recognize some patterns than to design a huge set of handcrafted rules for all possible pixel configurations. Traditionally, approaches using handcrafted visual features [72, 126] combined with a generic classifier or regressor [26] achieved satisfying results while making use of training data and prior information [108, 138]. With the advent of big data and highly powerful GPUs, deep neural networks (DNN) have managed to achieve human-level performance in many computer vision problems (image recognition [61, 114], image segmentation [102], full body pose estimation [19]). In general, convolutional layers play the role of a trainable feature extractor, while some fully connected layers do the regression/classification part. This kind of networks are called convolutional neural networks (CNNs or ConvNets) and can be trained end-to-end with stochastic gradient descent, as long as there are enough data. When dealing with ordered sequences, recurrent neural networks (RNN) provide a structure to model sequential dependencies. Besides computer vision, other domains use machine learning to interpret various types of data, and have benefited from the rise of deep learning. Examples include audio [121], natural language processing [25, 71], or even multi-modal *e.g.* audio-visual data [23].

Human robot interaction (HRI) is an interdisciplinary field. First, it features a robot, *i.e.* a mechanical system with sensors, able to perform a set of tasks. It needs to perceive and analyze sensor signals, take a decision and then move appropriately. Robotics is at the intersection of many fields: multimedia analysis (including computer vision), artificial intelligence, mechatronics, control theory, etc. Applications for robots include

industrial assemblies, intervention in hostile environment, domestic use, or as a companion. Most robots may or must interact with humans during normal activity for one reason or another. HRI addresses the problems that arise in this context, ranging from the design of robot-guiding interfaces to dealing with human intrusion into an autonomous robot working space [43]. In social HRI, interacting with people is the very purpose of the robotic system. Some social robots have already been used *e.g.* to help autistic children develop social skills [101] or as a bellboy in a hotel [93]. Social HRI uses results from psychophysics and neuroscience to model human behavior. Indeed, human-to-human interactions involve many complex, and often unconscious, communication processes that are very difficult for a robot to understand and accurately replicate.

In particular, gaze is a very prominent social cue. In a social interaction like a cocktail party or a meeting, it conveys a large quantity of information about relative social status, mental and emotional states, or interaction dynamics [18, 39]. Inferring gaze direction may often be necessary to predict the speech turn taking in a discussion [87]. In parallel, some non-social tasks can also benefit from gaze prediction. While driving a car, visual attention can help predict whether the driver should take a break [116]. On another note, determining the visual attention of people confronted to advertising banners or posters has business applications [103]. In this context, there exists a wide range of methods to infer gaze direction in miscellaneous contexts [47]. First, inferring eye gaze from an eye's patch image requires to take into account the variety of possible appearances. In general, eye-gaze methods work best when applied on a near frontal face. For this reason, head-mounted camera systems provide the most precise estimates of eye gaze [50]. Combined with a head-pose tracker, this provides reliable and consistent gaze estimations over time. However, for many social scenarios, setting up and calibrating a head-mounted system is impractical and cumbersome, and may even significantly impact people's behavior. There is a category of problems that cannot use a head-mounted system yet still manage to obtain near frontal faces: on-screen gaze following on smartphones and laptops. Embedded frontal cameras are designed to capture the face of the user watching the screen. The pervasiveness of camera-equipped devices has provided the opportunity for large-scale datasets [59, 139] and efficient algorithms [59, 139, 140]. Gaze-controlled computer interfaces have several applications *e.g.* for people with a motor disability [75], or more generally as a new tool in human-computer interface design [60]. Finally, in many unconstrained settings *e.g.* fixed cameras in a cocktail party, the head does not remain oriented towards the cameras and the eyes may even not be visible. Most work in this area use head orientation as an approximation of gaze direction [27, 83, 109]. Indeed, eyeball stays within 35° away from head orientation [118], and even less than that most of the time.

Gaze direction inference in itself is generally not the final objective. The decision process is better guided by the VFOA, *i.e.* who or what is being visually targeted. For example, a social robot must be able to make a difference between people roughly looking in the robot direction, and people that are purposively looking at the robot in order to interact with it [96]. Many works exist on the subject of VFOA inference [6, 32, 56, 76,

89, 110, 119, 135][1]. In more complex scenarios, *e.g.* as a museum guide [54], people jointly targeting an object provides a cue that the object is the topic of interest. The robot can decide to speak about the focused object of interest. In a multi-party discussion, like a stand-up conversation [56] or a meeting [6, 32, 89, 119], the participants VFOAs give strong insights on the conversation conduct. The focused person may be the one speaking or expected to speak next. A robot participating in the discussion can infer the speech-turn dynamics and possibly take the floor. The role of gaze in human-robot interaction is extensively discussed in [1].

Obviously, the VFOA cannot always be inferred directly. It requires to be aware that the target exists, and to know its location. When there is no prior information, a first strategy is to look for potential objects of interest. In a social interaction, people inherently are potential targets. Furthermore, finding regions that attract human's visual attention within the visual field is known as the saliency problem [33]. In parallel, gaze following consists in inferring the location of someone's VFOA. This skill appears during infancy and participates actively in development through joint attention [11, 17]. Recent works in computer vision combine gaze following and saliency [99, 100, 135] to simultaneously estimate the location of VFOA candidates and someone's actual VFOA. This bypasses the lack of prior information as long as both the person and his/her VFOA are within the visual field of view.

1.4 CONTRIBUTIONS

In this thesis, we address three successive problems that arise when a robot with limited field of view needs to estimate over time the Visual Focus of Attention of people involved in a social interaction, possibly including the robot itself. The problems can be described as follows. First, the robot needs a strategy to have and keep people in its field of view, since gaze is a visual cue. Second, it must locate the objects of interests; they may be too far away from people for the robot to see everything at once. Third, knowing the location of people and objects, the robot can start to guess people's VFOA on every camera frame. All these problems have been addressed with learning-based approaches.

In more detail, the contributions are the following:

- In order for the robot to keep people in its field of view, we propose a framework to learn a gaze control strategy based on audio-visual inputs. We use a recurrent neural network to map the sequence of audio and visual data into an action for moving the head. The network is trained using reinforcement learning (RL), with a reward based on the number of visible people. Hence, the robot autonomously learns to focus its attention to regions with multiple people, without the need for human supervision or external sensors. In addition to this framework, we introduce a synthetic environment that simulates a set of people moving and speaking to pre-train the model.

[1] Please note that the vocabulary in the literature is not standardized, some papers use for instance "gaze" or "eye-gaze" for VFOA

This avoids the need to spend hours in front of the robot for training. The model has been validated on the publicly available *AVDIAR* dataset, on our synthetic data, and on the Nao robot through transfer learning. Quantitative experiments allow to analyze the role of different hyper-parameters on the performance, as well as the respective importance of visual and auditory data. This work was done in common with Stéphane Lathuilière and led to the following publications:

  - Stéphane Lathuilière, Benoit Massé, Pablo Mesejo, and Radu Horaud. Neural network based reinforcement learning for audio-visual gaze control in human-robot interaction. *Pattern Recognition Letters*, 2018 [65],
  - Stéphane Lathuilière, Benoit Massé, Pablo Mesejo, and Radu Horaud. Deep reinforcement learning for audio-visual gaze control. In *IROS*, 2018 [64].

- People often look at an object that is not immediately visible to others. Yet, it is possible to approximately guess where this object stands. We propose a method to estimate the location of objects of interest solely based on the fact that they repeatedly are the target of someone's gaze direction. In practice, we use a heat-map embedding to represent the set of people's directions of interest, a *gaze heat-map*, and another one to represent the set of object locations, an *object heat-map*. This formulation allows us to take into account any number of people and objects. We combine the *gaze heat-maps* over time and train different versions of an encoder/decoder neural network to predict the *object heat-map*, as well as several baselines. The list of object locations is then retrieved by extracting local maxima on the *object heat-map*. We propose a synthetic data generator to have data diverse enough for training. The method has been validated on our synthetic data and, through transfer learning, on the publicly available *Vernissage* dataset. This work has not been published yet.

- In many social interactions, reacting appropriately often requires to know when someone changes his/her visual focus of attention. We propose to estimate and track jointly the VFOAs of a group of people over time, based on their head poses. To do so, we formulate a generative bayesian switching linear dynamical system that makes explicit the dependencies between head poses, gaze directions and VFOAs, as well as their dynamics. We address the inference problem on this model by extending the switching Kalman filter algorithm and propose a learning procedure for all parameters. All algorithms are computationally tractable. The method has been trained on the *Vernissage* dataset and tested on both the *Vernissage* dataset (using cross validation) and on the *LAEO* dataset. These two datasets are publicly available. This work led to the following publications:

  - Benoit Massé, Silèye Ba, and Radu Horaud. Simultaneous estimation of gaze direction and visual focus of attention for multi-person-to-robot interaction. In *IEEE ICME*, Seattle, WA, July 2016 [77],
  - Benoit Massé, Silèye Ba, and Radu Horaud. Tracking gaze and visual focus of attention of people involved in social interaction. *IEEE TPAMI*, 2017 [78],

It should be noted that, in this document, the contributions have been reported in reversed order since we believe it would help the reader better understand how they depend on one another.

1.5 RESOURCES

The work was performed in the Inria center in Grenoble [2], in the Perception team [3] under the supervision of Dr. Radu Horaud. This context helped me on several aspects. First, I benefited greatly from the co-supervision of Dr. Silèye Ba up to 2016, and later from Dr. Pablo Mesejo. It also gave me the opportunity to collaborate with Stéphane Lathuilière. Besides, the team has a lot of equipment to assist research on machine learning and robotics. A Nao robot [45] has been available for use, with engineers designing proper interfaces. The robot has been used in chapter 4. Other robots have been available in the team for a variety of uses but they will not be presented in this document. Additionally, there is a dummy head called Popeye, shown in Fig. 1.1(c). Popeye is a molded silicon reproduction of a human head on which are fixed a stereo camera and microphones. The goal is to mimic the acoustic properties of the human head.

The following datasets have been used in this thesis:

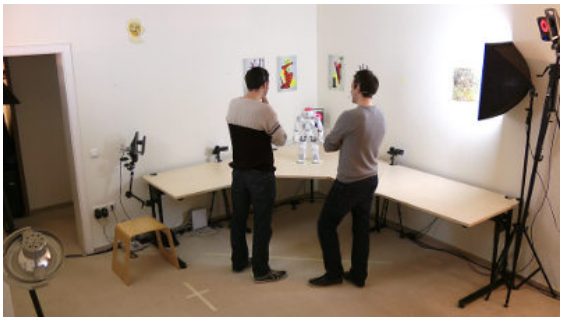
- The *Vernissage* dataset [54] is composed of ten recordings lasting ten minutes each. Each sequence contains two people interacting with a Nao robot and discussing about three wall paintings (see Fig. 1.1(a)). The robot plays the role of an art guide, describing the paintings and asking questions to the people in front of it. Each recording is split into two roughly equal parts. First, the robot describes the painting, leading to a one-way interaction. In the second part, the participants and the robot chat to answer a quiz. The scene was recorded with an RGB camera embedded into the robot head, and with a VICON motion capture system consisting of a network of infrared cameras. The VICON system providing accurate estimations for paintings' locations and for people and robot's head poses. Finally, the visual focus of attention of the participants are annotated over time. It has been used in chapters 2 and 3.

- The *LAEO* dataset [76] (dataset of people Looking At Each Other) is an extension of the *TVHID (TV Human Interaction Dataset)* [92]. It consists of 300 videos extracted from TV shows (see Fig. 1.1(b)). At least two actors appear in each video engaged in one of the following human-human interactions: handshake, highfive, hug, and kiss. There are 50 videos for each interaction and 100 videos with no interaction. *LAEO* is further annotated, namely some of these videos are split into shots which are separated by cuts. There are 443 shots in total, and each frame is manually annotated whether two people are looking at each other. All the faces in the dataset

[2] https://www.inria.fr/centre/grenoble
[3] https://team.inria.fr/perception/

are annotated with a bounding box and with a coarse head-orientation label: frontal-right, frontal-left, profile-right, profile-left, backward. It has been used in chapter 2.

- The *AVDIAR* dataset [40] (dataset for Audio-Visual Diarization) is a set of scenarios in which one to six people were asked to perform natural actions and speak. It is composed of dozens of videos with two high-resolution (1920×1080) video streams from a wide angle stereo camera, and six audio tracks. It was recorded in the team using the Popeye dummy head presented earlier, on which six microphones were fixed(see Fig. 1.1(c)). I personally participated in the making of this dataset, both in the recording and the annotation. It has been used in chapter 4.



(a) Sample from *Vernissage* dataset

(b) Sample from *LAEO* dataset

(c) sample from *AVDIAR* dataset, showing the Popeye dummy head

Figure 1.1: Illustration of the three datasets mentioned in this manuscript

Additionally, several GPUs with high computational power (Titan Xp, GTX 1070, etc.) were available to perform computationally expensive neural network training.

1.6 MANUSCRIPT STRUCTURE

This manuscript is organized as follow. In chapter 2, we propose a bayesian model to infer frame-by-frame visual focus of attention when the locations of objects of interest are known. In chapter 3, we present our method to estimate regions of interest from people gaze. In chapter 4, we describe our framework for training a robot to achieve an efficient gaze control strategy. Then, chapter 5 gives a perspective on the accomplished work, opens the discussion about some non-scientific aspects, and propose future developments.

In appendix A, the exhaustive set of training equations from chapter 2 are reported; then some details about the classes given during the PhD are presented in appendix B.

CHAPTER 2

TRACKING GAZE AND VISUAL FOCUS OF ATTENTION OF PEOPLE INVOLVED IN SOCIAL INTERACTION

2.1 INTRODUCTION

During a social interaction, people communicate by sending and receiving messages. Most explicit messages are transmitted through speech, but the communication process is facilitated by a large variety of non-verbal cues, *e.g.* prosody, hand gestures, body movements, head nodding, eye gaze, and facial expressions. In this chapter we are interested in estimating the *visual focus of attention* (VFOA), or who is looking at whom or at what, which has been recognized as one of the most prominent social cues. It is used in multi-party dialog to establish face-to-face communication, to respect social etiquette, to attract someone's attention, or to signify speech-turn taking, thus complementing speech communication.

The VFOA characterizes a perceiver/target pair. It is determined either by the line from the perceiver's face to the perceived target, or by the perceiver's *direction of sight* or *gaze direction* (which is often referred to as eye gaze or simply gaze). Indeed, one may state that the VFOA of person i is target j if the perceiver's gaze is aligned with the perceiver-to-target line. From a physiological point of view, eye gaze depends on both eyeball orientation and head orientation. Both the eye and the head are rigid bodies with three and six degrees of freedom respectively. The head location (three coordinates) and the head orientation (three angles) are jointly referred to as the *head pose*. With proper choices for the head- and eye-centered coordinate frames, one can assume that gaze is a combination of head pose and of eyeball orientation, and the VFOA depends on head pose, eyeball orientation, and target location.

In this chapter we are interested in estimating and tracking jointly the VFOAs of a group of people that communicate with each other and with a robot, or *multi-party* HRI

(human-robot interaction), which may well be viewed as a generalization of *single-user* HRI. From a methodological point of view the former is more complex than the latter. Indeed, in single-user HRI the person and the robot face each other and hence a camera mounted onto the robot head provides high-resolution frontal images of the user's face such that head pose and eye orientation can both be easily and robustly estimated. In the case of multi-party HRI the eyes are barely detected since the participants often turn their faces away from the camera. Consequently, VFOA estimation methods based on eye detection and eye tracking are ineffective and one has to estimate the VFOAs, indirectly, without explicit eye detection.

We propose a Bayesian switching dynamic model for the estimation and tracking gaze directions and VFOAs of several persons involved in social interaction. While it is assumed that head poses (location and orientation) and target locations can be directly detected from the data, the unknown gaze directions and VFOAs are treated as latent random variables. The proposed temporal graphical model, that incorporates gaze dynamics and VFOA transitions, yields (i) a tractable learning algorithm and (ii) an efficient gaze-and-VFOA tracking method.[1] The proposed method may well be viewed as a computational model of [37, 38]. The method is evaluated using two publicly available datasets, *Vernissage* [54] and *LAEO* [76]. These datasets consist of several hours of video containing situated dialog between two persons and a robot (*Vernissage*) and human-human interactions (*LAEO*). We are particularly interested in finding participants that either gaze to each other, gaze to the robot, or gaze to an object. *Vernissage* is recorded with a motion capture system (a network of infrared cameras) and with a camera placed onto the robot head. *LAEO* is collected from TV shows.

The remainder of this chapter is organized as follows. Section 2.2 provides an overview of related work in gaze, VFOA and head-pose estimation. Section 2.3 introduces the mathematical notations and definitions, states the problem formulation and describes the proposed model. Section 2.4 presents in detail the model inference and Section 2.5 derives the learning algorithm. Section 2.6 provides implementation details and Section 2.7 describes the experiments and reports the results.

2.2 Related Work

As already mentioned, the VFOA is correlated with gaze. Several methods proceed in two steps, in which the gaze direction is estimated first, and then used to estimate VFOA. In scenarios that rely on precise estimation of gaze [123, 133] a head-mounted camera, like the one in [50], can be used to detect the iris with high accuracy. Head-mounted eye trackers provide extremely accurate gaze measurements and in some circumstances eye-tracking data can be used to estimate objects of interest in videos [62]. Nevertheless, they are invasive instruments and hence not appropriate for analyzing social interactions.

[1] Supplementary materials are available at `https://team.inria.fr/perception/research/eye-gaze/`.

Gaze estimation is relevant for a number of scenarios, such as car driving [116] or interaction with smartphones [59]. In these situations, either the field of view is limited, hence the range of gaze directions is constrained (car driving), or active human participation ensures that the device yields frontal views of the user's face, thus providing accurate eye measurements [50, 79, 88, 116]. In some scenarios the user is even asked to limit head movements [73], or to proceed through a calibration phase [74, 88]. Even if no specific constraints are imposed, single-user scenarios inherently facilitate the task of eye measurement [79]. To the best of our knowledge, there is no gaze estimation method that can deal with unconstrained scenarios, *e.g.* participants not facing the cameras, partially or totally occluded eyes, etc. In general, eye analysis is inaccurate when participants are far away from the camera.

An alternative is to approximate gaze direction with head pose [85]. Unlike eye-based methods, head pose can be estimated from low-resolution images, *i.e.* distant cameras [20, 95, 98, 132, 137]. These methods estimate gaze only approximately since eyeball orientation can differ from head orientation by $\pm 35^\circ$ [118]. Gaze estimation from head orientation can benefit from the observation that gaze shifts are often achieved by synchronously moving the head and the eyes [37, 38, 44]. The correlation between head pose and gaze has also been exploited in [119]. More recently, [63] combined head and eye features to estimate the gaze direction using an RGB-D camera. The method still requires that both eyes are visible.

Several methods were proposed to infer VFOAs either from gaze directions [5], or from head poses [6, 76, 110, 135]. For example, in [76] it is proposed to build a gaze cone around the head orientation and targets lying inside this cone are used to estimate the VFOA. While this method was successfully applied to movies, its limitation resides in its vagueness: the VFOA information is limited to whether there are two people looking at each other or not.

An interesting application of VFOA estimation it the analysis of social behavior of participants engaged in meetings, *e.g.* [6, 32, 89, 119]. Meetings are characterized by interactions between seated people that interact based on speech and on head movements. Some methods estimate the most likely VFOA associated with a head orientation [89, 119]. The drawback of these approaches is that they must be purposively trained for each particular meeting layout. The correlation between VFOA and head pose was also investigated in [6] where an HMM is proposed to infer VFOAs from head and body orientations. This work was extended to deal with more complex scenarios, such as participants interacting with a robot [110, 111]. An input-output HMM is proposed in [111] to enable to model the following contextual information: participants tend to look to the speaker, to the robot, or to an object which is referred to by the speaker or by the robot. The results of [111] show that this improves the performance of VFOA estimation. Nevertheless, this method requires additional information, such as speaker identification or speech recognition.

The problem of joint estimation of gaze and of VFOA was addressed in a human-robot cooperation task [135]. In such a scenario the user doesn't necessarily face the camera and robot-mounted cameras have low-resolution, hence the estimation of gaze from direct

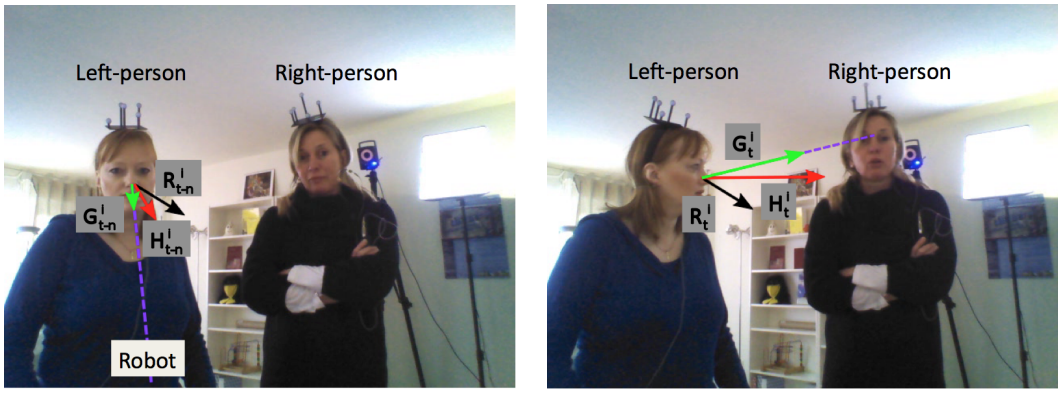


Figure 2.1: This figure illustrates the principle of our method and displays the observed and latent variables associated with a person (*left-person* indexed by i). The two images were grabbed with a camera mounted onto the head of a robot and they correspond to frames $t-n$ (left image) and t (right image), respectively. The following variables are displayed: head orientation (red arrow), $\mathbf{H}^i_{t-n}, \mathbf{H}^i_t$ (observed variables), as well as the latent variables estimated with the proposed method, namely gaze direction (green arrow), $\mathbf{G}^i_{t-n}, \mathbf{G}^i_t$, VFOA, $\mathrm{V}^i_{t-n}, \mathrm{V}^i_t$, and head reference orientation (black arrow), $\mathbf{R}^i_{t-n}, \mathbf{R}^i_t$ (that coincides with upper-body orientation). In this example *left-person* gazes towards the *robot* at $t-n$, then turns her head to eventually gaze towards *right-person* at t, hence her VFOA switches from $\mathrm{V}^i_{t-n} = \textit{robot}$ to $\mathrm{V}^i_t = \textit{right-person}$.

analysis of eye regions is not feasible. [135] proposes to learn a regression between the space of head poses and the space of gaze directions and then to predict an unknown gaze from an observed head pose. The head pose itself is estimated by fitting a 3D elliptical cylinder to a detected face, while the associated gaze direction corresponds to the 3D line joining the head center to the target center. This implies that during the learning stage, the user is instructed to gaze at targets lying on a table in order to provide training data. The regression parameters thus estimated correspond to a discrete set of head-pose/gaze-direction pairs (one for each target); an erroneous gaze may be predicted when the latter is not in the range of gaze directions used for training.

2.3 Proposed Model

The proposed mathematical model is inspired from psychophysics [37, 38]. In unconstrained scenarios a person switches his/her gaze from one target to another target, possibly using both head and eye movements. Quick eye movements towards a desired object of interest are called saccades. Eye movements can also be caused by the vestibule-ocular reflex that compensates for head movements such that one can maintain his/her gaze in the direction of the target of interest. Therefore, in the general case, gazing to an object is achieved by a combination of eye and head movements.

In the case of small gaze shifts, *e.g.* reading or watching TV, eye movements are predominant. In the case of large gaze shifts, often needed in social scenarios, head movements are necessary since eyeball movements have limited range, namely $\pm 35°$ [118].

Therefore, the proposed model considers that gaze shifts are produced by head movements that occur simultaneously with eye movements.

2.3.1 Problem Formulation

We consider a scenario composed of N active targets and M passive targets. An active target is likely to move and/or to have a leading role in an interaction. Active targets are persons and robots.[2] Passive targets are objects, *e.g.* wall paintings. The set of all targets is indexed from 0 to $N + M$, where the index 0 designates "no target". Let i be an active target (a person or a robot), $1 \leq i \leq N$, and j be a passive target (an object), $N + 1 \leq j \leq N + M$. A VFOA is a discrete random variable defined as follows: $\mathrm{V}_t^i = j$ means *person (or robot) i looks at target j at time t*. The VFOA of a person (or robot) i that looks at none of the known targets is $\mathrm{V}_t^i = 0$. The case $\mathrm{V}_t^i = i$ is excluded. The set of all VFOAs at time t is denoted by $\mathbf{V}_t = \left(\mathrm{V}_t^1, \ldots, \mathrm{V}_t^N\right)$.

Two continuous variables are now defined: head orientation and gaze direction. The head orientation of person i at t is denoted with $\mathbf{H}_t^i = [\phi_{H,t}^i, \theta_{H,t}^i]^\top$, *i.e.* the pan and tilt angles of the head with respect to some fixed coordinate frame. The gaze direction of person i is denoted with $\mathbf{G}_t^i$ and is also parameterized by pan and tilt with respect to the same coordinate frame, namely $\mathbf{G}_t^i = [\phi_{G,t}^i, \theta_{G,t}^i]^\top$. Although eyeball orientation is neither needed nor used, it is worth noticing that it is the difference between $\mathbf{G}_t^i$ and $\mathbf{H}_t^i$. These variables are illustrated on Fig. 2.1.

Finally, to establish a link between VFOAs and gaze directions, the target locations must be defined as well. Let $\mathbf{X}_t^i = [x_t^i, y_t^i, z_t^i]^\top$ be the location of target i. In the case of a person, this location corresponds to the head center while in the case of a passive target, it corresponds to the target center. These locations are defined in the same coordinate frame as above. Also notice that the direction from the active target i to target j is defined by the unit vector $\mathbf{X}_t^{ij} = (\mathbf{X}_t^j - \mathbf{X}_t^i)/\|\mathbf{X}_t^j - \mathbf{X}_t^i\|$ which can also be parameterized by two angles, $\mathbf{X}_t^{ij} = [\phi_{X,t}^{i,j}, \theta_{X,t}^{i,j}]^\top$.

As already mentioned, target locations and head orientations are observed random variables, while VFOAs and gaze directions are latent random variables. The problem to be solved can now be formulated as a maximum a posteriori (MAP) problem:

$$\hat{\mathbf{V}}_t, \hat{\mathbf{G}}_t = \underset{\mathbf{V}_t, \mathbf{G}_t}{\operatorname{argmax}} P(\mathbf{V}_t, \mathbf{G}_t | \mathbf{H}_{1:t}, \mathbf{X}_{1:t}). \tag{2.1}$$

Since there is no deterministic relationship between head orientation and gaze direction, we propose to model it probabilistically. For this purpose, we introduce an additional latent random variable, namely the head *reference* orientation, $\mathbf{R}_t^i = [\phi_{R,t}^i, \theta_{R,t}^i]^\top$, which we choose to coincide with the upper-body orientation. We use the following generative model, initially introduced in [6], linking gaze direction, head orientation, and head

[2] Note that in case of a robot, the gaze direction and the head orientation are identical and that the latter can be easily estimated from the head motors.

reference orientation:

$$P(\mathbf{H}_t^i|\mathbf{G}_t^i, \mathbf{R}_t^i; \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\mathbf{H}) = \mathcal{N}(\mathbf{H}_t^i; \boldsymbol{\mu}_{\mathbf{H},t}^i, \boldsymbol{\Sigma}_\mathbf{H}), \tag{2.2}$$

$$\text{with} \quad \boldsymbol{\mu}_{\mathbf{H},t}^i = \boldsymbol{\alpha}\mathbf{G}_t^i + (\mathbf{I}_2 - \boldsymbol{\alpha})\mathbf{R}_t^i, \tag{2.3}$$

where $\boldsymbol{\Sigma}_\mathbf{H} \in \mathbb{R}^{2\times2}$ is a covariance matrix, $\mathbf{I}_2 \in \mathbb{R}^{2\times2}$ is the identity matrix and $\boldsymbol{\alpha} = \text{Diag}\,(\alpha_1, \alpha_2)$ is a diagonal matrix of mixing coefficients, $0 < \alpha_1, \alpha_2 < 1$. Also it is assumed that the covariance matrix is the same for all the persons and over time. Therefore, head orientation is an observed random variable normally distributed around a convex combination between two latent variables: gaze direction and head reference orientation.

2.3.2 Gaze Dynamics

The following model is proposed:

$$P(\mathbf{G}_t^i|\mathbf{G}_{t-1}^i, \dot{\mathbf{G}}_{t-1}^i, \mathrm{V}_t^i = j, \mathbf{X}_t) = \mathcal{N}(\mathbf{G}_t^i; \boldsymbol{\mu}_{\mathbf{G},t}^{ij}, \boldsymbol{\Gamma}_\mathbf{G}), \tag{2.4}$$

$$P(\dot{\mathbf{G}}_t^i|\dot{\mathbf{G}}_{t-1}^i) = \mathcal{N}(\dot{\mathbf{G}}_t^i; \dot{\mathbf{G}}_{t-1}^i, \boldsymbol{\Gamma}_{\dot{\mathbf{G}}}), \tag{2.5}$$

with:

$$\boldsymbol{\mu}_{\mathbf{G},t}^{ij} = \begin{cases} \mathbf{G}_{t-1}^i + \dot{\mathbf{G}}_{t-1}^i\, dt, & \text{if } j = 0, \\ \boldsymbol{\beta}\mathbf{G}_{t-1}^i + (\mathbf{I}_2 - \boldsymbol{\beta})\mathbf{X}_t^{ij} + \dot{\mathbf{G}}_{t-1}^i\, dt, & \text{if } j \neq 0, \end{cases} \tag{2.6}$$

where $\dot{\mathbf{G}}_t^i = d\mathbf{G}_t^i/dt$ is the gaze velocity, $\boldsymbol{\Gamma}_\mathbf{G}, \boldsymbol{\Gamma}_{\dot{\mathbf{G}}} \in \mathbb{R}^{2\times2}$ are covariance matrices, and $\boldsymbol{\beta} = \text{Diag}\,(\beta_1, \beta_2)$ is a diagonal matrix of mixing coefficients, $0 < \beta_1, \beta_2 < 1$. Therefore, if a person looks at one of the targets, then his/her gaze dynamics depends on the person-to-target direction $\mathbf{X}_t^{ij}$ at a rate equal to $\boldsymbol{\beta}$, and if a person doesn't look at one of the targets, then his/her gaze dynamics follows a random walk.

The head reference orientation dynamics can be defined in a similar way:

$$P(\mathbf{R}_t^i|\mathbf{R}_{t-1}^i, \dot{\mathbf{R}}_{t-1}^i) = \mathcal{N}(\mathbf{R}_t^i; \boldsymbol{\mu}_{\mathbf{R},t}^i, \boldsymbol{\Gamma}_\mathbf{R}), \tag{2.7}$$

$$P(\dot{\mathbf{R}}_t^i|\dot{\mathbf{R}}_{t-1}^i) = \mathcal{N}(\dot{\mathbf{R}}_t^i; \dot{\mathbf{R}}_{t-1}^i, \boldsymbol{\Gamma}_{\dot{\mathbf{R}}}), \tag{2.8}$$

$$\text{with } \boldsymbol{\mu}_{\mathbf{R},t}^i = \mathbf{R}_{t-1}^i + \dot{\mathbf{R}}_{t-1}^i\, dt,$$

where $\dot{\mathbf{R}}_t^i = d\mathbf{R}_t^i/dt$ is the head reference orientation velocity and $\boldsymbol{\Gamma}_\mathbf{R}, \boldsymbol{\Gamma}_{\dot{\mathbf{R}}} \in \mathbb{R}^{2\times2}$ are covariance matrices. The dependencies between all the model variables are shown as a graphical representation in Fig. 2.2.

It is assumed that the gaze directions associated with different people are independent, given the VFOAs $\mathbf{V}_{1:t}$. The cross-dependency between different people is taken into account by the VFOA dynamics as detailed in section 2.3.3 below. Similarly, head orientations, and head reference orientations associated with different people are independent, given the VFOAs. By combining the above equations with this independence assumption,

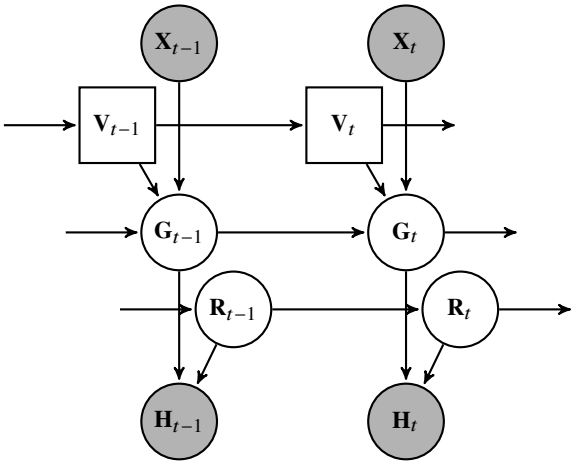


Figure 2.2: Graphical representation showing the dependencies between the variables of the proposed Bayesian dynamic model. The discrete latent variables (visual focus of attention) are shown with squares while continuous variables are shown with circles: observed variables (head poses and target locations) are shown with shaded circles and latent variables (gaze and reference directions) are shown with white circles.

we obtain:

$$P(\mathbf{H}_t|\mathbf{G}_t, \mathbf{R}_t) = \prod_i \mathcal{N}(\mathbf{H}_t^i; \boldsymbol{\mu}_{\mathbf{H},t}^i, \boldsymbol{\Sigma}_{\mathbf{H}}) \tag{2.9}$$

$$P(\mathbf{G}_t|\mathbf{G}_{t-1}, \dot{\mathbf{G}}_{t-1}, \mathbf{V}_t, \mathbf{X}_t) = \prod_{i,j} \mathcal{N}(\mathbf{G}_t^i; \boldsymbol{\mu}_{\mathbf{G},t}^{ij}, \boldsymbol{\Gamma}_{\mathbf{G}})^{\delta_j(\mathbf{V}_t^i)} \tag{2.10}$$

$$P(\mathbf{R}_t|\mathbf{R}_{t-1}, \dot{\mathbf{R}}_{t-1}) = \prod_i \mathcal{N}(\mathbf{R}_t^i; \boldsymbol{\mu}_{\mathbf{R},t}^i, \boldsymbol{\Gamma}_{\mathbf{R}}) \tag{2.11}$$

where the dependencies between variables are embedded in the variable means, *i.e.* (2.3) and (2.6). The covariance matrices will be estimated via training. While gaze directions can vary a lot, we assume that head reference orientations are almost constant over time, which can be enforced by imposing that the total variance of gaze is much larger than the total variance of head reference orientation, namely:

$$\mathrm{Tr}(\boldsymbol{\Gamma}_{\mathbf{G}}) \gg \mathrm{Tr}(\boldsymbol{\Gamma}_{\mathbf{R}}), \tag{2.12}$$

The trace of a covariance matrix is used here as an approximation of the variance for multidimensional variables. As the reference orientation is more stable than gaze direction, its variance is lower.

2.3.3 VFOA Dynamics

Using a first-order Markov approximation, the VFOA transition probabilities can be written as:

$$P(\mathbf{V}_t|\mathbf{V}_{1:t-1}) = P(\mathbf{V}_t|\mathbf{V}_{t-1}), \tag{2.13}$$

Notice that matrix $P(\mathbf{V}_t|\mathbf{V}_{t-1})$ is of size $(N+M)^N \times (N+M)^N$. Indeed, there are N persons (active targets), and $N + M + 1$ targets (one "no" target, N active targets and M passive targets) and the case of a person that looks to him/herself is excluded. For example, if $N = 2$ and $M = 4$, matrix (2.13) has $(2 + 4)^{2\times 2} = 1296$ entries. The estimation of this matrix would require, in principle, a large amount of training data, in particular in the presence of many symmetries. We show that, in practice, only 15 different transitions are possible. This can be seen on the following grounds.

We start by assuming conditional independence between the VFOAs at t:

$$P(\mathbf{V}_t|\mathbf{V}_{t-1}) = \prod_i P(\mathrm{V}_t^i|\mathbf{V}_{t-1}). \tag{2.14}$$

Let's consider V_t^i, the VFOA of person i at t, given $\mathbf{V}_{t-1}$, the VFOAs at $t-1$. One can distinguish two cases:

- $V_{t-1}^i = k$ where k is either a passive target, $N < k \leq N + M$, or it is none of the targets, $k = 0$; in this case V_t^i depends only on V_{t-1}^i, and
- $V_{t-1}^i = k$, where $k \neq i$ is a person $1 \leq k \leq N$; in this case V_t^i depends on the both V_{t-1}^i and V_{t-1}^k.

To summarize, we can write that:

$$P(\mathrm{V}_t^i = j|\mathbf{V}_{t-1}) = \begin{cases} P(\mathrm{V}_t^i = j|\mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = l) & \text{if } 1 \leq k \leq N, \\ P(\mathrm{V}_t^i = j|\mathrm{V}_{t-1}^i = k) & \text{otherwise.} \end{cases} \tag{2.15}$$

Additionally, we assume that, when switching VFOA to a new target without any special role, all such targets have an equal probability to be selected. Based on this, it is now possible to count the total number of possible VFOA transitions. With the same notations as in (2.15), we have the following possibilities:

- $k = 0$ (no target): there are two possible transitions, $j = 0$ and $j \neq 0$.
- $N < k \leq N + M$ (passive target): there are three possible transitions, $j = 0$, $j = k$, and $j \neq k$.
- $1 \leq k \leq N, l = 0$ (active target k looks at no target): there are three possible transitions, $j = 0$, $j = k$, and $j \neq k$.
- $1 \leq k \leq N, l = i$ (active target k looks at person i): there are three possible transitions, $j = 0$, $j = k$, and $j \neq k$.
- $1 \leq k \leq N, l \neq 0, i$ (active target k looks at active target l different than i): there are four possible transitions, $j = 0$, $j = k$, $j = l$ and $j \neq k, l$.

Therefore, there are 15 different possibilities for $P(\mathrm{V}_t^i = j|\mathbf{V}_{t-1})$, *i.e.* appendix A.1. Moreover, by assuming that the VFOA transitions don't depend on i, we conclude that the transition matrix may have up to 15 different entries. Moreover, the number of possible transitions is even smaller if there is no passive target ($M = 0$), or if the number of active targets is small, *e.g.* $N < 3$. This considerably simplifies the task of estimating this matrix and makes the task of learning tractable.

2.4 INFERENCE

We start by simplifying the notation, namely $\mathbf{L}_t = [\mathbf{G}_t; \dot{\mathbf{G}}_t; \mathbf{R}_t; \dot{\mathbf{R}}_t]$ where $[\cdot;\cdot]$ denotes vertical concatenation. The emission probabilities (2.9) become:

$$P(\mathbf{H}_t|\mathbf{L}_t) = \prod_i \mathcal{N}(\mathbf{H}_t^i; \boldsymbol{\mu}_{\mathbf{H},t}^i, \boldsymbol{\Sigma}_{\mathbf{H}}), \tag{2.16}$$

$$\text{with} \quad \boldsymbol{\mu}_{\mathbf{H},t}^i = \mathbf{C}\mathbf{L}_t^i, \tag{2.17}$$

where matrix $\mathbf{C}$ is obtained from the definition of $\mathbf{L}_t$ above and from (2.3):

$$\mathbf{C} = \begin{pmatrix} \alpha_1 & 0 & 0 & 0 & 1-\alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 & 0 & 1-\alpha_2 & 0 & 0 \end{pmatrix}.$$

The transition probabilities can be obtained by combining (2.10) and (2.11) with (2.5) and (2.8):

$$P(\mathbf{L}_t|\mathbf{V}_t, \mathbf{L}_{t-1}, \mathbf{X}_t) = \prod_i \prod_j \mathcal{N}(\mathbf{L}_t^i; \boldsymbol{\mu}_{\mathbf{L},t}^{ij}, \Gamma_{\mathbf{L}})^{\delta_j(\mathrm{V}_t^i)}, \tag{2.18}$$

$$\text{with} \quad \boldsymbol{\mu}_{\mathbf{L},t}^{ij} = \mathbf{A}_t^{ij}\mathbf{L}_{t-1}^i + \mathbf{b}_t^{ij} \tag{2.19}$$

$$\text{and} \quad \boldsymbol{\Gamma}_{\mathbf{L}} = \begin{pmatrix} \boldsymbol{\Gamma}_{\mathbf{G}} & & & \\ & \boldsymbol{\Gamma}_{\dot{\mathbf{G}}} & & \\ & & \boldsymbol{\Gamma}_{\mathbf{R}} & \\ & & & \boldsymbol{\Gamma}_{\dot{\mathbf{R}}} \end{pmatrix}, \tag{2.20}$$

where $\mathbf{A}_t^{ij}$ is an 8×8 matrix and $\mathbf{b}_t^{ij}$ is an 8×1 vector. The indices i,j and t cannot be dropped since the transitions depend on $\mathbf{X}_t^{ij}$ from (2.6).

The MAP problem (2.1) can now be derived in a Bayesian framework for the VFOA variables:

$$P(\mathbf{V}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) = \int P(\mathbf{V}_t, \mathbf{L}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) d\mathbf{L}_t. \tag{2.21}$$

We propose to study the filtering distribution of the joint latent variables, namely $P(\mathbf{V}_t, \mathbf{L}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t})$. Indeed, Bayes rule yields:

$$P(\mathbf{V}_t, \mathbf{L}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) \propto P(\mathbf{H}_t|\mathbf{L}_t) P(\mathbf{L}_t, \mathbf{V}_t|\mathbf{H}_{1:t-1}, \mathbf{X}_{1:t}). \tag{2.22}$$

Now we can introduce $\mathbf{V}_{t-1}$ and $\mathbf{L}_{t-1}$ using the sum rule:

$$\begin{aligned} P(\mathbf{L}_t, \mathbf{V}_t|\mathbf{H}_{1:t-1}, \mathbf{X}_{1:t}) &= \sum_{\mathbf{V}_{t-1}} \int P(\mathbf{L}_t, \mathbf{V}_t, \mathbf{L}_{t-1}, \mathbf{V}_{t-1}|\mathbf{H}_{1:t-1}, \mathbf{X}_{1:t}) d\mathbf{L}_{t-1} \\ &= \sum_{\mathbf{V}_{t-1}} \int P(\mathbf{L}_t|\mathbf{V}_t, \mathbf{L}_{t-1}, \mathbf{X}_t) P(\mathbf{V}_t|\mathbf{V}_{t-1}) \\ &\qquad \times P(\mathbf{L}_{t-1}, \mathbf{V}_{t-1}|\mathbf{H}_{1:t-1}, \mathbf{X}_{1:t-1}) d\mathbf{L}_{t-1}, \end{aligned} \tag{2.23}$$

where unnecessary dependencies were removed. Combining (2.22) and (2.23) we obtain a recursive formulation in $P(\mathbf{V}_t, \mathbf{L}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t})$. However, this model is still intractable without further assumptions. The main approximation used in this work consists of assuming local independence for the posteriors:

$$P(\mathbf{L}_t, \mathbf{V}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) \simeq \prod_i P(\mathbf{L}_t^i, \mathrm{V}_t^i|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}). \tag{2.24}$$

2.4.1 Switching Kalman Filter Approximation

Several strategies are possible, depending upon the structure of $P(\mathbf{L}_t, \mathbf{V}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t})$. Commonly used strategies to evaluate this distribution include variational Bayes or Monte-Carlo. Alternatively, we propose to cast the problem into the framework of switching Kalman filters (SKF) [84]. We assume the filtering distribution to be Gaussian,

$$P(\mathbf{L}_t, \mathbf{V}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) \propto \mathcal{N}(\mathbf{L}_t; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t). \tag{2.25}$$

From (2.24) and (2.25) we obtain the following factorization:

$$P(\mathbf{L}_t, \mathbf{V}_t|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) \propto \prod_i \prod_j \mathcal{N}(\mathbf{L}_t^i; \boldsymbol{\mu}_t^{ij}, \boldsymbol{\Sigma}_t^{ij})^{\delta_j(\mathrm{V}_t^i)}. \tag{2.26}$$

Thus, (2.23) can be split into N components, one for each active target i:

$$\begin{aligned} P(\mathbf{L}_t^i, \mathrm{V}_t^i = j|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) &\propto P(\mathbf{H}_t^i|\mathbf{L}_t^i) \\ &\quad \times \sum_{\mathbf{V}_{t-1}} \int \mathcal{N}(\mathbf{L}_t^i; A_t^{ij}\mathbf{L}_{t-1}^i + \mathbf{b}_t^{ij}) P(\mathrm{V}_t^i|\mathbf{V}_{t-1}) \\ &\quad \times \prod_k \mathcal{N}(\mathbf{L}_{t-1}^i; \boldsymbol{\mu}_{t-1}^{ik}, \boldsymbol{\Sigma}_{t-1}^{ik})^{\delta_k(\mathrm{V}_{t-1}^i)} d\mathbf{L}_{t-1}^i, \end{aligned} \tag{2.27}$$

or, after several algebraic manipulations:

$$P(\mathbf{L}_t^i, \mathrm{V}_t^i = j|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) \propto \sum_k w_{t-1,t}^{ijk} \mathcal{N}(\mathbf{L}_t^i; \boldsymbol{\mu}_t^{ijk}, \boldsymbol{\Sigma}_t^{ijk}). \tag{2.28}$$

In this expression, $\boldsymbol{\mu}_t^{ijk}$ and $\boldsymbol{\Sigma}_t^{ijk}$ are obtained by performing constrained Kalman filtering on $\boldsymbol{\mu}_{t-1}^{ik}$, $\boldsymbol{\Sigma}_{t-1}^{ik}$ with transition dynamics defined by $\mathbf{A}_t^{ij}$ and $\mathbf{b}_t^{ij}$, emission dynamics defined

by $\mathbf{C}$, and observation $\mathbf{H}_t^i$, *i.e.* [113]. The weights $w_{t-1,t}^{ijk}$ are defined as $P(\mathrm{V}_{t-1}^i = k \mid \mathrm{V}_t^i = j, \mathbf{H}_{1:t}, \mathbf{X}_{1:t})$. The constraint comes from the fact that

$$||\mathbf{G}_t^i - \mathbf{H}_t^i|| < 35^\circ \tag{2.29}$$

and is achieved by projecting the mean (refer to [113] for more details).

This can be rephrased as follows: from the filtering distribution at time $t-1$, there are $N+M$ possible dynamics for $\mathbf{L}_t^i$. The normal distribution at time $t-1$ then becomes a mixture of $N+M$ normal distributions at time t as shown in (2.28). However, we expect a single Gaussian such as $P(\mathbf{L}_t^i, \mathrm{V}_t^i = j|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}) \propto \mathcal{N}(\mathbf{L}_t^i; \boldsymbol{\mu}_t^{ij}, \boldsymbol{\Sigma}_t^{ij})$. This can be done by moment matching:

$$\boldsymbol{\mu}_t^{ij} = \sum_k w_{t-1,t}^{ijk} \boldsymbol{\mu}_t^{ijk} \tag{2.30}$$

$$\boldsymbol{\Sigma}_t^{ij} = \sum_k w_{t-1,t}^{ijk} (\boldsymbol{\Sigma}_t^{ijk} + (\boldsymbol{\mu}_t^{ijk} - \boldsymbol{\mu}_t^{ij})(\boldsymbol{\mu}_t^{ijk} - \boldsymbol{\mu}_t^{ij})^\top). \tag{2.31}$$

Finally, it is necessary to evaluate $w_{t-1,t}^{ijk}$. Let's introduce the following notations:

$$c_{t-1,t}^{ijk} = P(\mathrm{V}_t^i = j, \mathrm{V}_{t-1}^i = k|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}), \tag{2.32}$$

$$c_t^{ij} = P(\mathrm{V}_t^i = j|\mathbf{H}_{1:t}, \mathbf{X}_{1:t}). \tag{2.33}$$

It follows that

$$c_t^{ij} = \sum_k c_{t-1,t}^{ijk} \quad \text{and} \quad w_{t-1,t}^{ijk} = \frac{c_{t-1,t}^{ijk}}{c_t^{ij}}.$$

Applying Bayes formula to $c_{t-1,t}^{ijk}$ yields:

$$\begin{aligned} c_{t-1,t}^{ijk} \propto\ & P(\mathbf{H}_t|\mathrm{V}_t^i = j, \mathrm{V}_{t-1}^i = k, \mathbf{H}_{1:t-1}, \mathbf{X}_{1:t}) \\ & \times c_{t-1}^{ik} P(\mathrm{V}_t^i = j|\mathrm{V}_{t-1}^i = k, \mathbf{H}_{1:t-1}, \mathbf{X}_{1:t-1}). \end{aligned} \tag{2.34}$$

Then, c_{t-1}^{ik} is obtained from $c_{t-2,t-1}^{ijk}$ calculated at the previous time step. The last factor in (2.34) is either equal to $\sum_l c_{t-1}^{kl} P(\mathrm{V}_t^i = j|\mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = l)$ if k is an active target, or $P(\mathrm{V}_t^i = j|\mathrm{V}_{t-1}^i = k)$ otherwise. Both cases are straightforward to compute. Finally, the first factor in (2.34), the observation component, can be factorized as $P(\mathbf{H}_t^i|\mathrm{V}_t^i = j, \mathrm{V}_{t-1}^i = k, \mathbf{H}_{1:t-1}, \mathbf{X}_{1:t}) \times \prod_{n \neq i} \sum_m \sum_p P(\mathbf{H}_t^n|\mathrm{V}_t^n = m, \mathrm{V}_{t-1}^n = p, \mathbf{H}_{1:t-1}, \mathbf{X}_{1:t})$. By introducing the latent variable $\mathbf{L}$, we obtain:

$$\begin{aligned} & P(\mathbf{H}_t^n|\mathrm{V}_t^n = m, \mathrm{V}_{t-1}^n = p, \mathbf{H}_{1:t-1}, \mathbf{X}_{1:t}) \\ & = \int P(\mathbf{H}_t^n|\mathbf{L}_t^n)\ P(\mathbf{L}_t^n|\mathbf{L}_{t-1}^n, \mathrm{V}_t^n = m, \mathbf{X}_t) \\ & \times P(\mathbf{L}_{t-1}^n|\mathrm{V}_{t-1}^n = p, \mathbf{H}_{1:t-1}, \mathbf{X}_{1:t-1}) d\mathbf{L}_{t-1}^n d\mathbf{L}_t^n. \end{aligned} \tag{2.35}$$

All the factors in (2.35) are normal distributions, hence it integrates in closed-form. In summary, we devised a procedure to estimate an online approximation of the joint filtering distribution of the VFOAs, $\mathbf{V}_t$, and of the gaze and head reference directions, $\mathbf{L}_t$.

2.5 LEARNING

The proposed model has two sets of parameters that must be estimated: the transition probabilities associated with the discrete VFOA variables, and the parameters associated with the Gaussian distributions. Learning is carried out using Q recordings with annotated VFOAs. Each recording is composed of T_q frames, $1 \leq q \leq Q$ and contains N_q active targets (the robot is the active target 1 and the persons are indexed from 2 to N_q) and M_q passive targets. In addition to target locations and head poses, it is worth noting that the learning algorithm requires VFOA ground-truth annotations, while gaze directions are still treated as latent variables.

2.5.1 LEARNING THE VFOA TRANSITION PROBABILITIES

The VFOA transitions are drawn from the generalized Bernoulli distribution. Therefore, the transition probabilities can be estimated with $P(\mathrm{V}_t^i = j|\mathbf{V}_{t-1}) = \mathbb{E}_{t-1}[\delta_j(\mathrm{V}_t^i)]$, where $\delta_j(i)$ is the Kronecker delta function. In Section 2.3.3 we showed that there are up to 15 different possibilities for the VFOA transition probability. This enables us to derive an explicit formula for each case, see appendix A.2. Consider for example one of these cases, namely $p_{14} = P(\mathrm{V}_t^i = l|\mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = l)$, which is the conditional probability that at t person i looks at target l, given that at $t-1$ person i looked at person k and that person k looked at target l. This probability can be estimated with the following formula:

$$\hat{p}_{14} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_l(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}$$

2.5.2 LEARNING THE GAUSSIAN PARAMETERS

In Section 2.4 we described the derivation of the proposed model that is based on SKF. This model requires the parameters (means and covariances) of the Gaussian distributions defined in (2.16) and (2.18). Notice however that the mean (2.17) of (2.16) is parameterized by $\boldsymbol{\alpha}$. Similarly, the mean (2.19) of (2.18) is parameterized by $\boldsymbol{\beta}$. Consequently, the model parameters are:

$$\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}_{\mathbf{L}}, \boldsymbol{\Sigma}_{\mathbf{H}}), \tag{2.36}$$

and we remind that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are 2×2 diagonal matrices, $\boldsymbol{\Gamma}_{\mathbf{L}}$ is a 8×8 covariance and $\boldsymbol{\Sigma}_{\mathbf{H}}$ is a 2×2 covariance, and that we assumed that these matrices are common to all the active targets. Hence the total number of parameters is equal to $2 + 2 + 36 + 3 = 43$.

In the general case of SKF models, the discrete variables are unobserved both for learning and for inference. Here we propose a learning algorithm that takes advantage of

the fact that the discrete variables, *i.e.* VFOAs, are observed during the learning process, namely the VFOAs are annotated. We propose an EM algorithm adapted from [14]. In the case of a standard Kalman filter, an EM iteration alternates between a forward-backward pass to compute the expected latent variables (E-step), and between the maximization of the expected complete-data log-likelihood (M-step).

We start by describing the M-step. The complete-data log-likelihood is:

$$\begin{aligned}\ln P(\mathbf{L}^1, \mathbf{H}^1, \ldots, \mathbf{L}^Q, \mathbf{H}^Q|\boldsymbol{\theta}) \\ = \sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q} \ln P(\mathbf{L}_t^{q,i}|\mathbf{L}_{t-1}^{q,i}, \boldsymbol{\beta}, \boldsymbol{\Gamma}_{\mathbf{L}}) \\ + \sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=1}^{T_q} \ln P(\mathbf{H}_t^{q,i}|\mathbf{L}_t^{q,i}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_{\mathbf{H}}).\end{aligned} \tag{2.37}$$

By taking the expectation w.r.t. the posterior distribution $P(\mathbf{L}^1, \ldots, \mathbf{L}^Q|\mathbf{H}^1, \ldots, \mathbf{H}^Q, \boldsymbol{\theta})$, we obtain:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \mathbb{E}_{\mathbf{L}^1,\ldots,\mathbf{L}^K|\boldsymbol{\theta}^{\text{old}}}\left[\ln P(\mathbf{L}^1, \mathbf{H}^1, \ldots, \mathbf{L}^Q, \mathbf{H}^Q|\boldsymbol{\theta})\right], \tag{2.38}$$

which can be maximized w.r.t. to the parameters $\boldsymbol{\theta}$, which yields closed-form expressions for the covariance matrices

$$\boldsymbol{\Gamma}_{\mathbf{L}} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q} \mathbb{E}[(\mathbf{L}_t^{q,i} - \boldsymbol{\mu}_{\mathbf{L},t}^{q,ij})(\mathbf{L}_t^{i} - \boldsymbol{\mu}_{\mathbf{L},t}^{q,ij})^\top]}{\sum_{q=1}^{Q}(N_q - 1)(T_q - 1)}, \tag{2.39}$$

where $\boldsymbol{\mu}_{\mathbf{L},t}^{q,ij} = \mathbf{A}_t^{q,ij}\mathbf{L}_{t-1}^{q,i} + \mathbf{b}_t^{q,ij}$, *i.e.* (2.19), and:

$$\boldsymbol{\Sigma}_{\mathbf{H}} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=1}^{T_q} \mathbb{E}[(\mathbf{H}_t^{q,i} - \boldsymbol{\mu}_{\mathbf{H},t}^{q,i})(\mathbf{H}_t^{q,i} - \boldsymbol{\mu}_{\mathbf{H},t}^{q,i})^\top]}{\sum_{q=1}^{Q}(N_q - 1)T_q}, \tag{2.40}$$

where $\boldsymbol{\mu}_{\mathbf{H},t}^{q,i} = \mathbf{C}\mathbf{L}_t^{q,i}$, *i.e.* (2.17).

The estimation of $\boldsymbol{\alpha}$ and of $\boldsymbol{\beta}$ is carried out in the following way. $\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})/\partial\beta_1 = 0$

and $\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})/\partial\beta_2 = 0$ yield a set of two linear equations in the two unknowns for $\boldsymbol{\beta}$

$$\begin{aligned}\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\mathbb{E}\left[(\mathbf{L}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{L},t}^{q,ij})^\top\boldsymbol{\Gamma}_{\mathbf{L}}^{-1}\frac{\partial}{\partial\beta_1}(\mathbf{L}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{L},t}^{q,ij})\right]=0,\\ \sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\mathbb{E}\left[(\mathbf{L}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{L},t}^{q,ij})^\top\boldsymbol{\Gamma}_{\mathbf{L}}^{-1}\frac{\partial}{\partial\beta_2}(\mathbf{L}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{L},t}^{q,ij})\right]=0,\end{aligned} \tag{2.41}$$

and similarly: for $\boldsymbol{\alpha}$

$$\begin{aligned}\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=1}^{T_q}\mathbb{E}\left[(\mathbf{H}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{H},t}^{q,i})^\top\boldsymbol{\Sigma}_{\mathbf{H}}^{-1}\frac{\partial}{\partial\alpha_1}(\mathbf{H}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{H},t}^{q,i})\right]=0,\\ \sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=1}^{T_q}\mathbb{E}\left[(\mathbf{H}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{H},t}^{q,i})^\top\boldsymbol{\Sigma}_{\mathbf{H}}^{-1}\frac{\partial}{\partial\alpha_2}(\mathbf{H}_t^{q,i}-\boldsymbol{\mu}_{\mathbf{H},t}^{q,i})\right]=0,\end{aligned} \tag{2.42}$$

where as above, the expectation is taken w.r.t. to the posterior distribution. Once the formulas above are expanded and once the means $\boldsymbol{\mu}_{\mathbf{L},t}^{q,ij}$ and $\boldsymbol{\mu}_{\mathbf{H},t}^{q,i}$ are substituted with their expressions, the following terms remain to be estimated: $\mathbb{E}[\mathbf{L}_t^{q,i}]$, $\mathbb{E}[\mathbf{L}_t^{q,i}\mathbf{L}_t^{q,i\top}]$ and $\mathbb{E}[\mathbf{L}_t^{q,i}\mathbf{L}_{t-1}^{q,i\top}]$.

The E-step provides estimates of these expectations via a forward-backward algorithm. For the sake of clarity, we drop the superscripts i (active target index) and q (recording index) up to equation (2.49) below. Introducing the notation $P(\mathbf{L}_t|\mathbf{H}_1,\ldots,\mathbf{H}_t) = \mathcal{N}(\mathbf{L}_t;\boldsymbol{\mu}_t,\mathbf{P}_t)$, the forward-pass equations are:

$$\boldsymbol{\mu}_t = \mathbf{A}_t\boldsymbol{\mu}_{t-1} + \mathbf{b}_t + \mathbf{K}_t(\mathbf{H}_t - \mathbf{C}(\mathbf{A}_t\boldsymbol{\mu}_{t-1} + \mathbf{b}_t)) \tag{2.43}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t,t-1}, \tag{2.44}$$

where

$$\mathbf{P}_{t,t-1} = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^\top + \boldsymbol{\Gamma}_{\mathbf{L}}, \tag{2.45}$$

$$\mathbf{K}_t = \mathbf{P}_{t,t-1}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{t,t-1}\mathbf{C}^\top + \boldsymbol{\Sigma}_{\mathbf{H}})^{-1}. \tag{2.46}$$

The backward pass estimates $P(\mathbf{L}_t|\mathbf{H}_1,\ldots,\mathbf{H}_T) = \mathcal{N}(\mathbf{L}_t;\hat{\boldsymbol{\mu}}_t,\hat{\mathbf{P}}_t)$ and leads to

$$\hat{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_t + \mathbf{J}_t(\hat{\boldsymbol{\mu}}_{t+1} - (\mathbf{A}_{t+1}\boldsymbol{\mu}_t + \mathbf{b}_{t+1})), \tag{2.47}$$

$$\hat{\mathbf{P}}_t = \mathbf{P}_t + \mathbf{J}_t(\hat{\mathbf{P}}_{t+1} - \mathbf{P}_{t+1,t})\mathbf{J}_t^\top, \tag{2.48}$$

where

$$\mathbf{J}_t = \mathbf{P}_t\mathbf{A}_{t+1}^\top(\mathbf{P}_{t+1,t})^{-1}. \tag{2.49}$$

The expectations are estimated by performing a forward-backward pass over all the persons and all the recordings of the training data. This yields the following formulas:

$$\mathbb{E}[\mathbf{L}_t^{q,i}] = \hat{\boldsymbol{\mu}}_t^{q,i} \tag{2.50}$$

$$\mathbb{E}[\mathbf{L}_t^{q,i}\mathbf{L}_t^{q,i\top}] = \hat{\mathbf{P}}_t^{q,i} + \hat{\boldsymbol{\mu}}_t^{q,i}\hat{\boldsymbol{\mu}}_t^{q,i\top} \tag{2.51}$$

$$\mathbb{E}[\mathbf{L}_t^{q,i}\mathbf{L}_{t-1}^{q,i\top}] = \hat{\mathbf{P}}_t^{q,i}\mathbf{J}_{t-1}^{q,i\top} + \hat{\boldsymbol{\mu}}_t^{q,i}\hat{\boldsymbol{\mu}}_{t-1}^{q,i\top} \tag{2.52}$$

2.6 Implementation Details

The proposed method was evaluated on the *Vernissage* dataset [54] and on the *Looking At Each Other (LAEO)* dataset [76]. We describe in detail these datasets and their annotations. We provide implementation details and we analyse the complexity of the proposed algorithm.

2.6.1 The *Vernissage* Dataset

The *Vernissage* scenario can be briefly described as follows, *e.g.* Fig. 2.3: there are three wall paintings, namely the passive targets denoted with o_1, o_2, and o_3 ($M = 3$); two persons, denoted left person (left-p) and right person (right-p), interact with the robot, hence $N = 3$. The robot plays the role of an art guide, describing the paintings and asking questions to the two persons in front of him. Each recording is split into two roughly equal parts. The first part is dedicated to painting explanation, with a one-way interaction. The second part consists of a quiz, thus illustrating a dialog between the two participants and the robot, most of the time concerning the paintings.

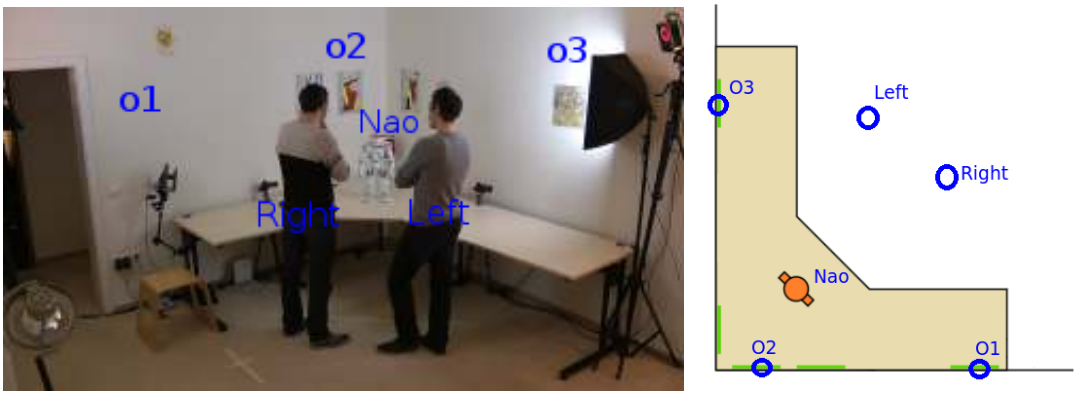


Figure 2.3: The *Vernissage* setup. Left: Global view of an "exhibition" showing wall paintings, two participants, *i.e.* left-p and right-p, and the NAO robot. Right: Top view of the room showing the *Vernissage* layout.

The scene was recorded with a camera embedded into the robot head and with a VICON motion capture system consisting of a network of infrared cameras, placed onto

the walls, and of optical markers, placed onto the robot and people heads. Both were recorded at 25 frames per second (fps). There is a total of ten recordings, each lasting ten minutes. The VICON system provided accurate estimates of head locations, $\overline{\mathbf{X}}_{1:T}$ and head orientations, $\overline{\mathbf{H}}_{1:T}$. Head locations and head orientations were also estimated using from the RGB images gathered with the camera embedded into the robot head. The RGB images are processed as follows. We use the OpenCV version of [126] to detect faces and their bounding boxes which are then tracked over time using [9]. Next, we extract HOG descriptors from each bounding box and apply the head orientation estimator from [31]. This yields $\tilde{\mathbf{H}}_{1:T}$. The 3D head locations, $\tilde{\mathbf{X}}_{1:T}$, can be estimated using the line of sight through the face center and the bounding-box size, which provides a rough estimate of the depth along the line of sight.

In the remaining of this chapter, $\overline{\mathbf{X}}_{1:T}$ and $\overline{\mathbf{H}}_{1:T}$ are referred to as *Vicon Data*; $\tilde{\mathbf{X}}_{1:T}$ and $\tilde{\mathbf{H}}_{1:T}$ as *RGB Data*. Because the whole setup was carefully calibrated, both Vicon and RGB Data are represented in the same coordinate frame.

In all our experiments we assumed that the passive targets are static and their locations are provided in advance. The location of the robot itself is also known in advance and one can easily estimate the orientation of the robot head from motor readings. Finally, the VFOAs of the participants were manually annotated in all the frames of all the recordings.

2.6.2 THE *LAEO* DATASET

The *LAEO* dataset [76] is an extension of the *TVHID (TV Human Interaction Dataset)* [92]. It consists of 300 videos extracted from TV shows. At least two actors appear in each video engaged in four human-human interactions: handshake, highfive, hug, and kiss. There are 50 videos for each interaction and 100 videos with no interaction. The videos have been grabbed at 25 fps and each video lasts from five seconds to twenty-five seconds. *LAEO* is further annotated, namely some of these videos are split into shots which are separated by cuts. There are 443 shots in total which are manually annotated whenever two persons look at each other, [76].

While there is no passive target in this dataset ($M = 0$), the number of active targets (N) corresponds to the number of persons in each shot. In practice N varies from one to eight persons. All the faces in the dataset are annotated with a bounding box and with a coarse head-orientation label: frontal-right, frontal-left, profile-right, profile-left, backward. As with *Vernissage* , we use the bounding-box center and size to estimate the 3D coordinates of the heads, $\overline{\mathbf{X}}_{1:T}$. We assigned a pan angle to each one of the five coarse head orientations, $\overline{\mathbf{H}}_{1:T}$. We also computed finer head orientations, $\tilde{\mathbf{H}}_{1:T}$, using [31].

2.6.3 ALGORITHMIC DETAILS

The INFERENCE procedure is summarized in Algorithm 1. This is basically an iterative filtering procedure. The UPDATE step consists of applying the recursive relationship, derived in Section 2.4, to $\boldsymbol{\mu}_t^{ij}$, $\boldsymbol{\Sigma}_t^{ij}$ and c_t^{ij}, using $\boldsymbol{\mu}_t^{ijk}$, $\boldsymbol{\Sigma}_t^{ijk}$ and $c_{t-1,t}^{ijk}$ as intermediate

variables. The VFOA is chosen using MAP, given observations up to the current frame, and the gaze direction is the mean of the filtered distribution (the first two components of $\boldsymbol{\mu}_t^{ij}$ are indeed the mean for the pan and tilt gaze angles).

Data: $\mathbf{X}_{1:T}$, $\mathbf{H}_{1:T}$
Result: $\mathbf{G}_{1:T}$, $\mathbf{V}_{1:T}$

$c_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1 \leftarrow$ INITIALIZATION$(\mathbf{H}_1, \mathbf{X}_1)$
for $i = 1..N$ **do**
  $\mathrm{V}_1^i \leftarrow \mathrm{argmax}_j\, c_1^{ij}$
  $\mathbf{G}_1^i \leftarrow \boldsymbol{\mu}_1^{ij}[1..2]$
end
for $t = 2..T$ **do**
  $c_t, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t \leftarrow$ UPDATE$(\mathbf{H}_t, \mathbf{X}_t, c_{t-1}, \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$
  for $i = 1..N$ **do**
    $\mathrm{V}_t^i \leftarrow \mathrm{argmax}_j\, c_t^{ij}$
    $\mathbf{G}_t^i \leftarrow \boldsymbol{\mu}_t^{ij}[1..2]$
  end
end

Algorithm 1: INFERENCE

Let's now describe the INITIALIZATION procedure used by Algorithm 2. In a probabilistic framework, parameter intialization is generally addressed by defining an initial distribution, *e.g.* $P(\mathbf{L}_1|\mathbf{V}_1)$. Here, we did not explicitly define such a distribution. Initialization is based on the fact that, with repeated similar observation inputs, the algorithm reaches a steady-state. The initialization algorithm uses a repeated update method with initial observation to provide an estimate of gaze and of reference directions. Consequently, the initial filtering distribution $P(\mathbf{L}_1, \mathbf{V}_1|\mathbf{H}_1, \mathbf{X}_1)$ is implicitly defined as the expected stationary state.

Data: $\mathbf{X}_1$, $\mathbf{H}_1$
Result: $c_{init}, \boldsymbol{\mu}_{init}, \boldsymbol{\Sigma}_{init}$

$\boldsymbol{\mu}_{init} \leftarrow [\mathbf{H}_1; \mathbf{0}; \mathbf{H}_1; \mathbf{0}]$
$\boldsymbol{\Sigma}_{init} \leftarrow \mathbf{I}$
$c_{init} \leftarrow \frac{1}{N+M}$; `// Uniform`
while *Not Convergence* **do**
  $c_{init}, \boldsymbol{\mu}_{init}, \boldsymbol{\Sigma}_{init} \leftarrow$ UPDATE$(\mathbf{H}_1, \mathbf{X}_1, c_{init}, \boldsymbol{\mu}_{init}, \boldsymbol{\Sigma}_{init})$
end

Algorithm 2: INITIALIZATION

2.6.4 Algorithm Complexity

The computational complexity of Algorithm 1 is

$$C = (T + T_I)C_U, \tag{2.53}$$

where T is the number of frames in a test video, T_I is the number of iterations needed by the Algorithm 2 (initialization) to converge and C_U is the computational complexity of UPDATE. Let's detail the cost of C_U. From Section 2.4 one sees that the following values need to be computed: $P(\mathbf{H}_t^i|\mathrm{V}_t^i = j, \mathrm{V}_{t-1}^i = k, \mathbf{H}_{1:t-1}, \mathbf{X}_{1:t-1})$, $c_{t-1,t}^{ijk}$, $\boldsymbol{\mu}_t^{ijk}$, $\boldsymbol{\Sigma}_t^{ijk}$, and then c_t^{ij}, $\boldsymbol{\mu}_t^{ij}$ and $\boldsymbol{\Sigma}_t^{ij}$, for each active target i, and for each combination of targets j and k different from i. There are N possible values for i and $(N + M)$ possible values for j and k. Then,

$$C_U = K \times N(N + M)^2, \tag{2.54}$$

where K is a factor whose complexity can be estimated as follows. The most time-consuming part is the Kalman Filter algorithm used to estimate $\boldsymbol{\mu}_t^{ijk}$ and $\boldsymbol{\Sigma}_t^{ijk}$ from $\boldsymbol{\mu}_t^{ik}$ and $\boldsymbol{\Sigma}_t^{ik}$. These calculations are dominated by several 8×8 and 2×8 matrix inversions and multiplications. By neglecting scalar multiplications and matrix additions, and by denoting with C_{KF} the complexity of the Kalman filter, we obtain that $K \approx C_{KF}$ and hence $C_U \approx C_{KF} \times N(N + M)^2$.

Additionally, the way the algorithm has been designed makes it possible to be used online, *i.e.* receiving observations $\mathbf{H}_t$ and $\mathbf{X}_t$ one by one. In that case, we are interested in the computational cost of one iteration of algorithm 1

$$C_t = C_U + C_O, \quad \textit{if} \quad t > 1 \tag{2.55}$$

where C_O is the cost required to obtain observations. It mostly consists in head pose detection and tracking. For most recent algorithms that rely on deep learning architecture, $C_O \gg C_U$.

2.7 Experimental results

2.7.1 *Vernissage* Dataset

We applied the same experimental protocol to the Vicon and RGB data. We used a *leave-one-video-out* strategy for training. The test is performed on the left out video. We used the frame recognition rate (FRR) metrics to quantitatively evaluate the methods. FRR is person-wise and corresponds to the percentage of frames for which his/her VFOA is correctly estimated. One should note however that the ground-truth VFOAs were obtained by manually annotating each frame in the data. This is subject to errors since the annotator has to associate a target with each person.

The VFOA transition probabilities and the model parameters were estimated using the learning method described in Section 2.5. Appendix A.2 provides the formulas used

Table 2.1: FRR scores of the estimated VFOAs for the Vicon data for the left and right persons (left-p and right-p).

Recording	Ba & Odobez [6]		Proposed	
	left-p	right-p	left-p	right-p
09	51.6	**65.1**	**59.8**	61.4
10	64.3	**74.4**	**76.5**	65.0
12	53.5	**67.6**	**61.6**	63.2
15	**67.1**	46.2	64.8	**67.6**
18	37.5	28.3	**62.0**	**53.7**
19	**56.7**	45.4	54.5	**60.4**
24	44.9	49.0	**59.7**	**54.7**
26	40.3	32.9	**43.6**	**43.1**
27	65.8	72.0	**79.8**	**78.3**
30	69.1	49.1	**72.0**	**63.9**
Mean	54.5		**62.6**	

for estimating the VFOA transition probabilities given the annotated data. Note that the fifteen transitions probabilities thus estimated are identical for both data, Vicon and RGB.

The Gaussian parameters, *i.e.* (2.36), were estimated using the EM algorithm of Section 2.5.2. This learning procedure requires head-pose estimates as well as the targets locations, estimated as just explained. Since these estimates are different for the two kinds of data (Vicon and RGB) we carried out the learning process twice, with the Vicon data and with the RGB data. The EM algorithm needs initialization. The initial parameter values for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are $\boldsymbol{\alpha}^0 = \boldsymbol{\beta}^0 = \text{Diag}\,(0.5, 0.5)$. Matrices $\boldsymbol{\Sigma}_{\mathbf{H}}$ and $\boldsymbol{\Gamma}_{\mathbf{L}}$ defined in (2.20) are initialized with isotropic covariances: $\boldsymbol{\Sigma}_{\mathbf{H}}^0 = \sigma\mathbf{I}_2$, $\boldsymbol{\Gamma}_{\mathbf{G}}^0 = \boldsymbol{\Gamma}_{\dot{\mathbf{G}}}^0 = \gamma\mathbf{I}_2$, and $\boldsymbol{\Gamma}_{\mathbf{R}}^0 = \boldsymbol{\Gamma}_{\dot{\mathbf{R}}}^0 = \eta\mathbf{I}_2$ with $\sigma = 15$, $\gamma = 5$, and $\eta = 0.5$. In particular, this initialization is consistent with (2.12). In practice we noticed that the covariances estimated by training remain consistent with (2.12).

2.7.2 Results with Vicon Data

The FRR of the estimated VFOAs for the Vicon data are summarized in Table 2.1. A few examples are shown in Fig. 2.5. The FRR score varies between 28.3% and 74.4% for [6] and between 43.1% and 79.8% for the proposed method. Notice that high scores are obtained by both methods for recording #27. Similarly, low scores are obtained for recording #26. Since both methods assume that head motions and gaze shifts occur synchronously, an explanation could be that this hypothesis is only valid for some of the participants. The confusion matrices for VFOA classification using Vicon data are given in Fig. 2.4. There are a few similarities between the results obtained with the two methods. In particular, wall painting $\#o_2$ stands just behind Nao and both methods don't always discriminate between these two targets. In addition, the head of one of the persons is often aligned with painting $\#o_1$ from the viewpoint of the other person. A similar remark holds for painting

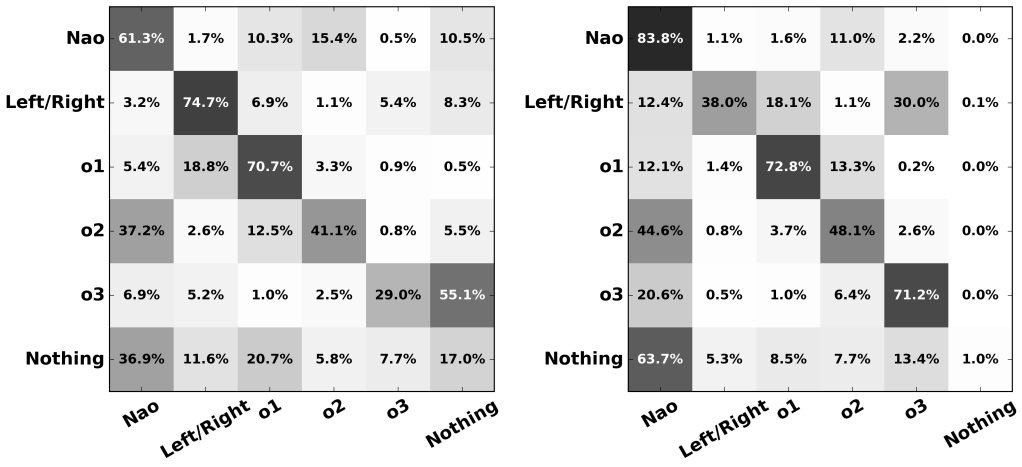


Figure 2.4: Confusion matrices for the Vicon data. Left: [6]. Right: Proposed algorithm. Row-wise: ground-truth VFOAs. Column-wise: estimated VFOAs. Diagonal terms represent the recall.

$\#o_3$. As a consequence both methods often confuse the VFOA in these cases. This can be seen in the third image of Fig. 2.5. Indeed, it is difficult to estimate whether the left person (left-p) looks at $\#o_1$ or at *right-p*.

Finally, both methods have problems with recognizing the VFOA "nothing" or gaze aversion ($V_t^i = 0$). We propose the following explanation: the targets are widespread in the scene, hence it is likely that an acceptable target lies in most of the gaze directions. Moreover, Nao is centrally located, therefore the head orientation used to look at Nao is similar to the resting head orientation used for gaze aversion. However, in [6] the reference head orientation is fixed and poorly suited for dynamic head-to-gaze mapping, hence the high error rate on painting $\#o_3$. Our method favors the selection of a target, either active or passive, over the no target (nothing) case.

2.7.3 Results with RGB Data

The RGB images were processed as described in section 2.6.1 above in order to obtain head orientations, $\tilde{\mathbf{H}}_{1:T}$, and 3D head locations, $\tilde{\mathbf{X}}_{1:T}$. Table 2.2 shows the accuracy of these measurements (in degrees and in centimeters), when compared with the ground truth provided by the VICON motion capture system. As it can be seen, while the head orientation estimates are quite accurate, the error in estimating the head locations can be as large as 0.8 m for participants lying in between 1.5 m and 2.5 m in front of a robot, *e.g.* recordings #19 and #24. In particular this error increases as a participant is farther away from the robot. In these cases, the bounding box is larger than it should be and hence the head location is, on an average, one meter closer than the true location. These relatively large errors in 3D head location affect the overall behavior of the algorithm.

The FRR scores obtained with the RGB data are shown in Table 2.3. As expected the loss in accuracy is correlated with the head location error: the results obtained with

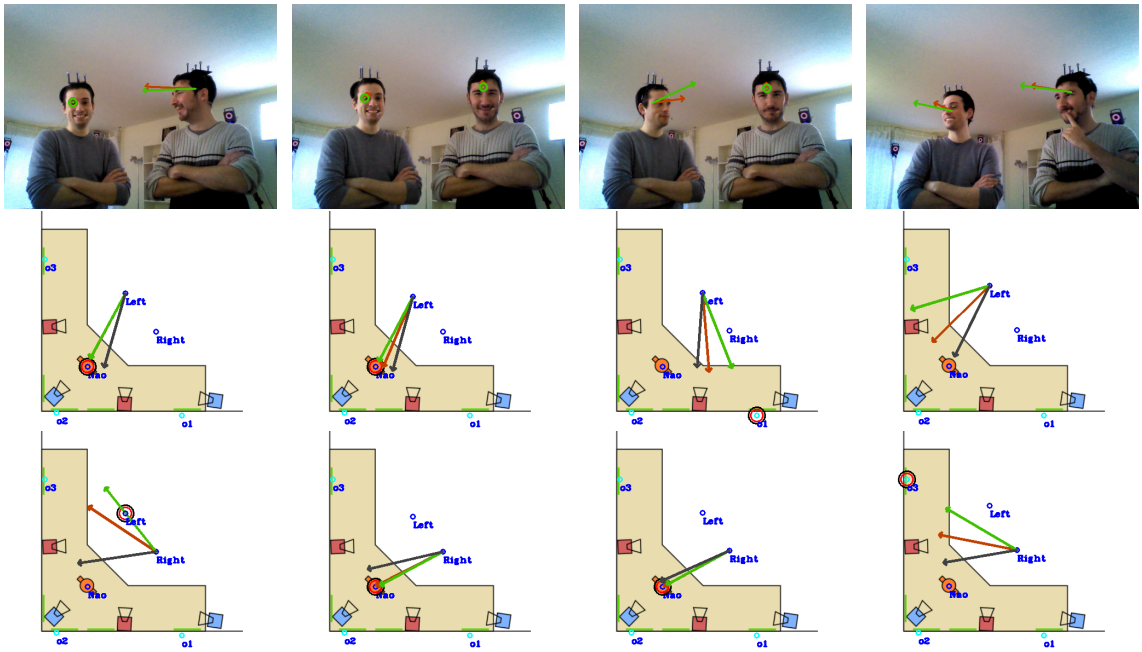


Figure 2.5: Results obtained with the proposed method on Vicon data. Gaze directions are shown with green arrows, head reference directions with dark-grey arrows and observed head directions with red arrows. The ground-truth VFOA is shown with a black circle. The top row displays the image of the robot-head camera. Top views of the room show results obtained for the left-p (middle row) and for the right-p (bottom row). In the last example the left-p gazes at "nothing".

recordings #09 and #30 are close to the ones obtained with the Vicon data, whereas there is a significant loss in accuracy for the other recordings. The loss is notable for [6] in the case of the right person (right-p) for the recordings #12, #18 and #27. The confusion

Table 2.2: Mean error for head pose estimations from RGB data, for the left person (left-p) and the right person (right-p). The errors in head location (centimeters) and orientation (degrees) are computed with respect to values provided by the motion capture system. Recordings #10 and #15 have been omitted because some annotations are missing and the comparison would be biased.

Video	Location error (cm)		Pan error		Tilt error	
	left-p	right-p	left-p	right-p	left-p	right-p
09	18.1	20.8	4.4°	4.8°	3.7°	3.2°
12	35.7	41.5	4.8°	5.5°	2.6°	3.8°
18	36.9	12.8	6.8°	3.7°	5.8°	2.5°
19	86.0	87.4	4.0°	5.8°	2.7°	3.7°
24	86.5	73.9	3.3°	3.5°	2.8°	2.7°
26	50.2	56.9	7.4°	9.0°	4.1°	5.2°
27	64.5	58.3	4.1°	5.8°	3.2°	4.4°
30	16.7	13.3	2.8°	2.9°	1.8°	2.7°
Mean	46.4		5.0°		3.3°	

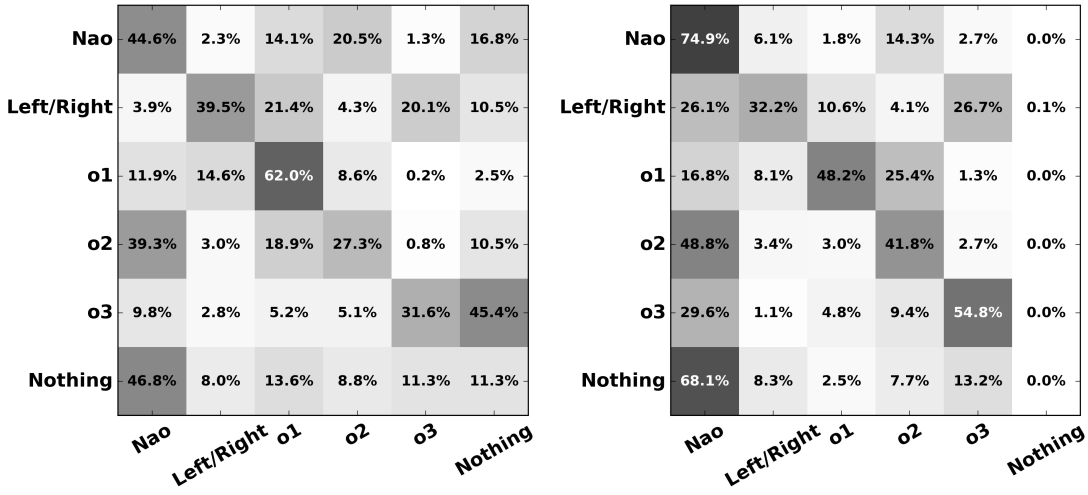


Figure 2.6: Confusion matrices for the RGB data. Left: [6]. Right: Proposed algorithm. Row-wise: ground-truth VFOAs. Column-wise: estimated VFOAs. Diagonal terms represent the recall.

matrices obtained with the RGB data are shown on Fig. 2.6.

In the case of RGB data, the comparison between our method and the method of [111] is biased by the use of different head orientation and 3D head location estimators. Indeed, the RGB data results reported in [111] were obtained with unpublished methods for estimating head orientations and 3D head locations, and for head tracking. Moreover, [111] uses cross-modal information, namely the speaker identity based on the audio track (one of the participants or the robot) as well as the identity of the object of interest. We also note that [111] reports mean FRR values obtained over all the test recordings, instead of an FRR value for each recording. Table 2.4 summarizes a comparison between the average FRR obtained with our method, with [6], and with [111]. Our method yields a similar FRR score as [111] using the Vicon data (first row) in which case the same head pose inputs are used.

Table 2.3: FRR scores of the estimated VFOAs obtained with [6] and with the proposed method for the RGB data. The last two columns show the 3D head location errors of Table 2.2 for reminder. The high reported errors in location or pan estimation from recordings #19, #24 and #26 led to an ongoing inconsistency of the geometric model; they have been ignored in the evaluation.

Video	Ba & Odobez [6]		Proposed		Head pos. error	
	left-p	right-p	left-p	right-p	left-p	right-p
09	50.3	**59.8**	**58.1**	55.9	18.1	20.8
12	54.2	14.8	**59.0**	**46.5**	35.7	41.5
18	39.0	16.1	**64.2**	**33.1**	36.9	12.8
27	38.2	17.1	**53.3**	**55.1**	64.5	58.3
30	**61.6**	44.6	54.7	**66.6**	16.7	13.3
Mean	39.0		**54.7**			

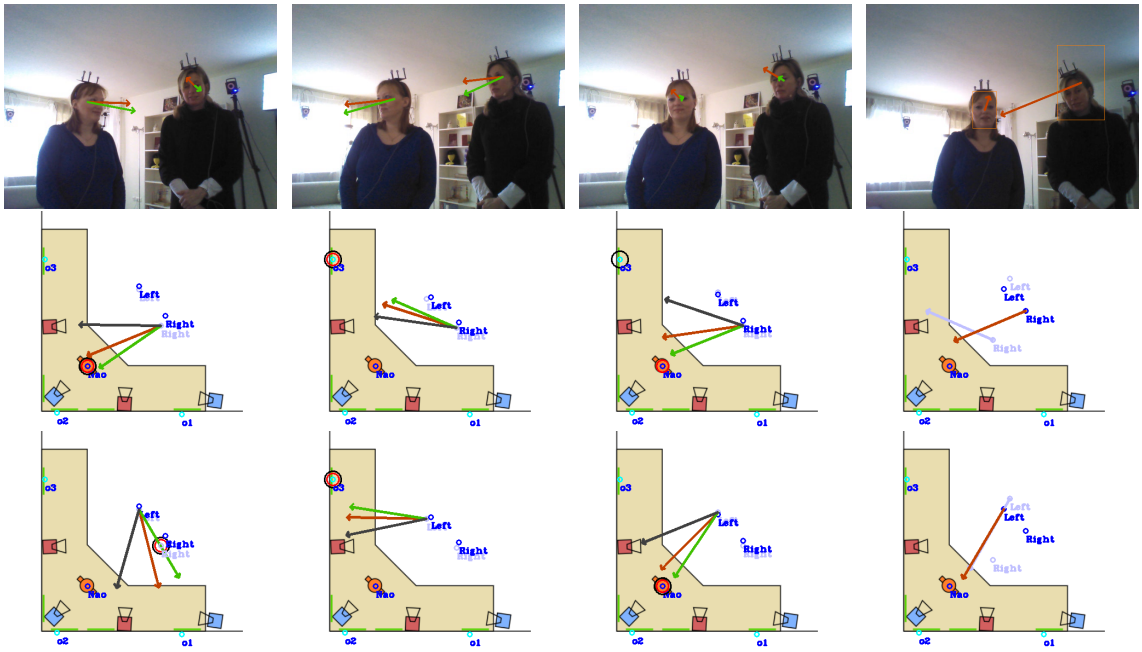


Figure 2.7: Results obtained with the proposed method on RGB data. Gaze directions are shown with green arrows, head reference directions with dark-grey arrows and observed head directions with red arrows. The ground-truth VFOA is shown with a black circle. The top row displays the image of the robot-head camera. Top views of the room show results obtained for the left person (left-p, middle row) and the right person (right-p, bottom row).

Table 2.4: Mean FRR scores obtained with [6], with [111] and with the proposed method. Recording #26 was excluded from the FRR means as reported in [111]. Moreover, [111] uses additional contextual information.

	Ba & Odobez [6]	Sheikhi [111]	Proposed
Vicon data	56.5	**66.6**	64.7
RGB data	39.0	**62.4**	54.7

2.7.4 LAEO DATASET

As already mentioned in Section 2.6.2 above, the *LAEO* annotations are incomplete to estimate the person-wise VFOA at each frame. Indeed, the only VFOA-related annotation is whether two people are looking at each other during the shot. This is not sufficient to know in which frames they are actually looking at each other. Moreover, when more than two people appear in a shot, the annotations don't specify who are the people that look at each other. For these reasons, we decided to estimate the parameters using Vicon data of the whole *Vernissage* dataset, *i.e.* cross-validation.

We used the same pipeline as with the *Vernissage* RGB data to estimate 3D head locations, $\tilde{\mathbf{X}}_{1:T}$, from the face bounding boxes. Concerning head orientation, there are two cases: coarse head orientations (manually annotated) and fine head orientations (estimated). Coarse head orientations were obtained in the following way: pan and tilt values

Figure 2.8: This figure shows some results obtained with the *LAEO* dataset. The top row shows results obtained with coarse head orientation and the bottom row shows results obtained with fine head orientation. Head orientations are shown with red arrows. The algorithm infers gaze directions (green arrows) and VFOAs (blue circles). People looking at each others are shown with a dashed blue line.

were associated with each head orientation label, namely the pan angles $-20°$, $20°$, $-80°$, $80°$, and $180°$ were assigned to labels *frontal-left*, *frontal-right*, *profile-left*, *profile-right*, and *backwards* respectively, while a tilt angle of $0°$ was assigned to all five labels. Fine head orientations were estimated using the same procedure as in the case of the *Vernissage* RGB data, namely face detection, face tracking, and head orientation estimation using [31]. Algorithm 1 was used to compute the VFOA for each frame and for each person thus allowing to determine who looks at whom, *e.g.* Fig. 2.8.

We used two shot-wise, not frame-wise, metrics since the *LAEO* annotations are for each shot: the *shot recognition rate* (SRR), *e.g.* Table 2.5, and the *average precision* (AP), *e.g.* Table 2.6. We note that [76] only provides AP scores. It is interesting to note that the proposed method yields results comparable with those of [76] on this dataset. This is quite remarkable knowing that we estimated the model parameters with the *Vernissage* training data.

Table 2.5: Average shot recognition rate (SRR) obtained with [6] and with the proposed method.

	Ba & Odobez [6]	Proposed
Coarse head orientation	0.535	**0.727**
Fine head orientation	0.363	**0.479**

Table 2.6: Average precision (AP) obtained with [76], with Ba & Odobez [6] and with the proposed method.

	Marin-Jimenez et al. [76]	[6]	Proposed
Coarse head orientation	**0.925**	0.916	0.923
Fine head orientation	**0.896**	0.838	0.890

2.8 CONCLUSIONS

In this chapter we addressed the problem of estimating and tracking gaze and visual focus of attention of a group of participants involved in social interaction. We proposed a Bayesian state-space model that exploits the correlation between head movements and eye gaze on one side, and between visual focus of attention and eye gaze on the other side. We described in detail the proposed formulation. In particular we showed that the entries of the large-sized matrix of VFOA transition probabilities have a very small number of different possibilities for which we provided closed-form formulae. The immediate consequence of this simplified transition matrix is that the associated learning doesn't require a large training dataset. We showed that the problem of simultaneously inferring VFOAs and gaze directions over time can be cast in the framework of a switching Kalman filter which, in our case, yields tractable learning and inferring algorithms.

We applied the proposed method to two datasets, *Vernissage* and *LAEO*. *Vernissage* contains several recordings of a human-robot interaction scenario. We experimented both with motion capture data gathered with a VICON system and with RGB data gathered with a camera mounted onto a robot head. We also experimented with the *LAEO* dataset that contains several hundreds of video shots extracted from TV shows. A quite remarkable result is that the parameters obtained by training the model with the *Vernissage* data have been successfully used for testing the method with the *LAEO* data, *i.e.* cross-validation. This can be explained by the fact that social interactions, even in different contexts, share a lot of characteristics. We compared our method with three other methods, based on HMMs [6], on input-output HMMs [111], and on a geometric model [76]. The interest of these methods (including ours) resides in the fact that eye detection, unlike many existing gaze estimation methods, is not needed. This feature makes the above methods practical and effective in a very large number of situations, *e.g.* social interaction.

Our method however has several limitations. First, the model is not robust to errors in the premises. Indeed, when people look at an object which existence is unknown, the method can at best output "unknown" (this case happens a lot with the *LAEO* dataset), but may also be flexible enough to consistently mistake the associated VFOA. Overcoming this limitation requires the possibility to dynamically change the set of objects. The next chapter will provide some clues for future research in this direction. Second, the VFOA transitions are sometimes guided by unobtainable data, *e.g.* the subject of the discussion. At last, we can note that gaze inference from head orientation is an ill-posed problem. Indeed, the correlation between gaze and head movements is person dependent as well as context dependent. It is however important to infer gaze whenever the eyes cannot be reliably observed in images and properly analyzed. We proposed to solve the problem based on the fact that alignments often occur between gaze directions and several targets, which is a sensible assumption in practice.

Contextual information could considerably improve the results. Indeed, additional information such as speaker recognition (as in [111]), speaker localization [67], speech recognition, or speech-turn detection [40] may be used to learn VFOA transitions in

multi-party multimodal dialog systems. In parallel, with the recent breakthroughs in deep learning, we would soon expect to be able to directly estimate eye-gaze in quite difficult conditions, leading to more robust models. Finally, one could argue that an adapted neural network architecture may be more efficient in capturing the complex dynamics of gaze and visual focus of attention. We will see in the next chapter a deep learning formulation including dependencies between gaze direction, head orientation and VFOA. A deep learning formulation for this chapter could be based on the same ideas. However, mixing such a formulation with the expert knowledge introduced in the current probabilistic model is a difficult problem.

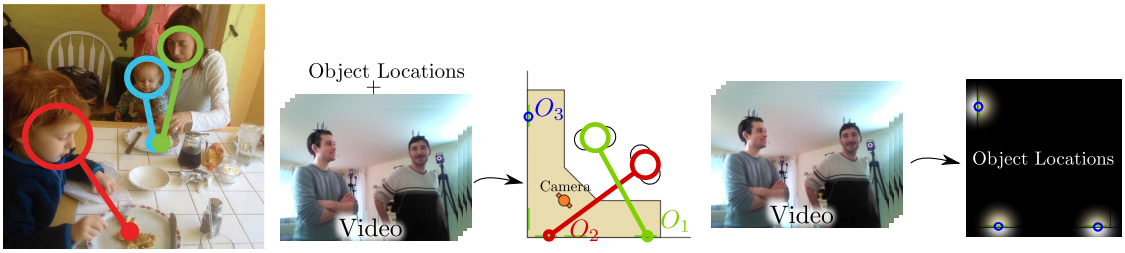
CHAPTER 3

UNCONSTRAINED GAZE-FOLLOWING IN VIDEOS: DETECTION OF OUT-OF-VIEW OBJECTS

3.1 INTRODUCTION

Humans have the ability to estimate where other people are looking at by following their direction of gaze, and to infer which object or person they are looking at. For example, in the Vernissage dataset, the VFOAs are correctly annotated, although they lie outside the camera field of view most of the time. In chapter 2, we proposed a method to continuously estimate people's VFOAs, under the assumption that we know the location of each object of interest. In many scenarios, there is no reason to have this information a priori. However, humans are capable of intuiting the focused region of space from the gaze direction of an observer. This skill is called *gaze following*.

Gaze following is used very early in human development; infants use it to achieve joint attention and quickly improve learning language [11]. More generally, it helps understanding the actions other people are performing in a scene and, even, predict what they might do next. An accurate estimation of where one or several persons look has an enormous potential in order to determine which are the objects of interest in a scene, predict the actions and movements of the participants and, in general terms, advance towards a better visual scene understanding. Depending on the application, this skill can be used to build an empirical saliency map, *i.e.* a probability map of each region to be looked at during a specific scenario. It can also be used to analyze the social behavior of a group and, in the case of a robot, exploit its knowledge to decide an interaction strategy. In the case of a personal-assistant robot, this ability could enable the robot to adequately respond to simple requests, such as "Give me the house keys", simply by looking at the keys and without explicitly pointing to them or providing additional information. Previous works, e.g. [15] have also highlighted that gaze direction is a strong attentional cue



(a) Gaze detection [99] **(b)** Visual Focus of Attention **(c)** Unconstrained Gaze-following

Figure 3.1: A comparison of gaze-related computer vision problems. In the standard formulation of gaze-following (a), the problem consists in localizing the objects that people are likely to be looking at (and both observer and objects are visible in the input image). Visual focus of attention estimation (b) consists in associating which person is looking at what object at a certain moment (considering that the objects locations are known). In unconstrained gaze-following (or visual regions of attention detection) (c), we aim at localizing objects of interest even if they are not visible in the video image.

in guiding eye movements, complementing low-level saliency cues, and have concluded that it should be considered in constructing more predictive visual attention models.

This chapter addresses the detection of visual regions of attention, which are expected to contain objects of interest. People in a video generally either look at other people or at an object of interest. Such an object can be indistinctly located inside or outside the current image. In the standard *gaze-following* problem, addressed *e.g.* in [99], both the observer and the targeted object are within the same image. An example is provided on Fig. 3.1(a). This is significantly different from the problem of estimating the *visual focus of attention* from chapter 2 in which object locations are known, but potentially non-visible (occluded or outside the field of view, see Fig. 3.1(b)). Both formulations ignore that an object may not be visible within the image, and its location is most probably unknown in an unconstrained scenario. All the more in a social interaction, an object is not "of interest" until people actually start paying attention to it. In this chapter, we deal with *unconstrained gaze-following* on videos, e.g. Fig 3.1(c), meaning that we tackle the more general problem of predicting the location of objects of interest whose number and locations are not known a priori, and that are not necessarily visible.

Our method takes as input a video sequence containing a group of people, and outputs a set of estimated locations for the objects of interest. This work makes the assumption that objects do not move across the video sequence. As in chapter 2, we propose to use the head orientation as a strong cue for the gaze direction. The pipeline, illustrated in Fig. 3.2, is as follow. First, from each video image, a bounding box for each detected face is extracted. The location and orientation of each face is computed in a fixed global system. Head pose (orientation and location) information are combined to obtain a top-view representation of the scene encoded in a set of heat-map embeddings. We feed that information to a convolutional neural network based on an encoder/decoder architecture to estimate a probability heat-map, *i.e.* a grid containing the likelihood for each region of space that an object of interest is located here. Finally, the set of object locations is obtained by performing local maxima detection on the probability heat-map.

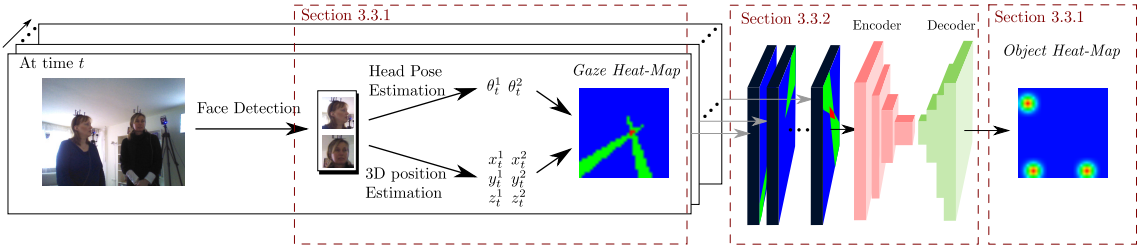


Figure 3.2: Outline of the proposed model. For every frame and detected face, orientation and 3D location are estimated, and both sources of information are combined to obtain a top-view representation of the scene encoded in a heat-map. The sequence of heat-maps is then given to a neural network with an encoder/decoder architecture. The network outputs a heat-map that predicts the position of the objects of interest in the top-view domain.

The reasons for using heat-map embeddings are multiple. Indeed, the exact number of people and objects is not known a priori and may vary within and between video sequences. Heat-map structures are independent of the number of participants (people and objects of interest). Additionally, the problem addressed is fundamentally geometric, and heat-maps intrinsically encode the geometry of the scene. Moreover, convolutional neural networks are able to efficiently extract this structured information in order to obtain a descriptive input representation. A drawback of the heat-map representation is the difficulty to predict an object outside the modeled area. Nevertheless, for indoor scenarios, the area containing the objects is bounded. It is then possible to adapt the heat-map size for the current setup and train the model using scaled simulated scenarios (see Section 3.4). For all these reasons, we decided to use heat-map embeddings.

The contribution of this chapter is threefold. First, we propose a novel formalism for embedding the spatial representation of directions of interest and regions of attention. They are modeled as a top-view heat-map, *i.e.* a discrete grid of spatial regions from a top-view perspective. Contrary to previous work, this formalism is not limited to representing locations within the field of view. Second, we propose several different convolutional encoder/decoder neural network architectures that learn to predict object locations from head poses in the heat-map domain. Third, since a large amount of data is required to train a deep neural network, we propose an algorithm based on a generative probabilistic framework that can sample an unlimited number of synthetic conversational scenarios, involving people and objects of interest.

The remainder of this chapter is organized as follows. A state of the art is presented in Section 3.2. Then, the details of the proposed heat-map representations and neural network architectures are respectively given in Sections 3.3.1 and 3.3.2. The synthetic data generation process is described in Section 3.4. Section 3.5 is dedicated to experimental results, both on synthetic and real data. To conclude, Section 3.6 discusses the perspectives and limitations of this work.

3.2 RELATED WORK

Finding objects of interest generally requires to analyze the visual field of view and look for highly contrasting regions. Indeed, an object or a person is likely to look different from the background, thus highly contrasting regions have higher chance of containing something interesting. This approach, similar to the human brain pipeline [124], is studied in computer vision under the term *saliency* [53]. A salient region is one that attracts the visual attention of an observer. There is a large community working on proposing efficient saliency models [48, 70, 90, 105, 127]. Moreover, when the objects of interest are expected to have a particular appearance *e.g.* faces, it is possible to run an adapted detector [66, 126].

However, *saliency* is different from *gaze-following* [34]. Indeed, since the point of view of people inside the image is different from the camera (and from each other), there is a discrepancy between fields of view. The problem of gaze-following overlaps with VFOA. In both cases, the goal is to match the gaze of a person with a location containing a region of interest. The difference lies in the fact that the potential regions of interest are known a priori. Indeed, the locations are obtained through a separate process, using external sensor, manual annotations, or an adapted detector [6, 32, 76, 89, 111, 119]. In chapter 2, we did the experiments using face information both from a face detector and from annotated data. Nevertheless, in the *Vernissage* dataset, the paintings never are within the field of view of the robot camera, and we had to rely on the annotated data. As opposed to these works, in gaze-following, the goal is to estimate the location of objects of interest. It is neither known a priori, nor is it delegated to an external module.

Gaze-following requires to find regions that are salient *i.e.* that attract gaze, from another point of view. However, a salient region is most likely salient from most points of view. Based on this remark, [99] combines a saliency model with a gaze direction model to find salient objects at the intersection of the image and the person's field of view. The attention predictor in [129] also uses both saliency and gaze. By combining multiple gaze directions, [35] estimates shared attention of multiple people, but still within the image. In [100], the authors further investigate this problem based on the idea that the gaze target of a person inside a video may be visible in another video frame. Their method still relies on a saliency model. Finally, [29, 107] merge the problems of saliency and gaze-following in the context of human-robot interactions. Indeed, the robot is both an active member of the scenario, and an observer behind the camera. Both papers are based on saliency and gaze direction, as well as additional data such as pointing gesture and speech. However, all works based on saliency require that the object of interest lies within the field of view. By contrast, we wish to be able to locate out-of-view objects; we cannot rely on this category of methods.

Apart from saliency-based gaze-following, a few other methods have been published, addressing the gaze following problem in the 3D space instead of the image plane. [117] proposes to estimate 3D regions of attention using only the location of people. They model social group structures that constrain the set of candidate locations. In this framework, they learn to locate regions of attention independently of visual saliency. Their

method only needs people locations and can work in complex scenarios, using only spatial data from first person cameras. However, it fails when some people are undetected and the group structures are wrongly estimated, or when a person is isolated and should not be integrated into a group structure. By contrast, both [24] and [16] independently propose to use 3D intersection of gazes in a probabilistic framework to estimate locations of objects of interest, possibly outside the camera field of view. The methods achieve good levels of performance – even though [24] lacks quantitative evaluation. In both cases, no training data have been used. Each method is designed with strong geometric assumptions so that location inference can be performed without any prior learning phase. At the time this thesis was written, the data on which the methods have been tested were not released yet for comparison.

In this chapter, we combine a learning-based model with a geometric formulation to address the gaze-following problem, without the restriction of being limited to the image plan. Only few works exist in this direction [24, 117], and use strong social or geometric assumptions.

3.3 Deep Learning for Unconstrained Gaze-Following

We note N_t the number of persons at time $t \in \{1 \dots T\}$. For each person, we suppose that we can estimate its corresponding 3D head location $\mathbf{X}_t^n = [x_t^n, y_t^n, z_t^n]^\top, n \in \{1 \dots N_t\}$, and head orientation $\mathbf{H}_t^n = [\phi_{H,t}^n, \theta_{H,t}^n]^\top$ in a common scene-centered coordinate frame. This fits the modeling used in chapter 2. However, we additionally choose to drop the z-coordinate (the height) and the head tilt angle as in [24], projecting every object and every person in the same horizontal plane. As we will see later, this simplification drastically reduces the complexity of the model while still representing plausible scenarios. In the remaining of the chapter, the term *position* refers to 2D coordinates $\mathbf{x}_t^n = [x_t^n, y_t^n]^\top$ in the horizontal plane (top-view perspective), and *head orientation* refers to the head pan angle $\phi_{H,t}^n$, abbreviated as ϕ_t^n from now on.

3.3.1 Heat-Map Representation

We employ heat-map representations of the scene from a top-view perspective. The scene is discretized into a 2D grid of dimension $S_U \times S_V$. Therefore, each position in the scene $\mathbf{x} = (x, y)$ is associated to a grid cell $\mathbf{p} = (u, v) \in \{1 \dots S_U\} \times \{1 \dots S_V\}$. As stated previously, $\mathbf{x}$ is bounded in both dimensions: $x \in [x_{min}, x_{max}]$ and $y \in [y_{min}, y_{max}]$. With these notations, $\mathbf{p} = (u, v)$ is obtained from $\mathbf{x}$ as

$$\begin{cases} u &= \lceil S_U \times \frac{x - x_{min}}{x_{max} - x_{min}} \rceil \\ v &= \lceil S_V \times \frac{y - y_{min}}{y_{max} - y_{min}} \rceil \end{cases} \tag{3.1}$$

where $\lceil \cdot \rceil$ is the *ceiling* function. The grid cell associated to $\mathbf{x}_t^n = (x_t^n, y_t^n)$, the position of a person n at time t, is written $\mathbf{p}_t^n$.

In this formalism, a heat-map $\mathbf{\Lambda}$ is a 2D map of $S_U \times S_V$ elements that attaches to each cell $\mathbf{p}$ of the grid a value $\mathbf{\Lambda}(\mathbf{p})$ between 0 and 1. The meaning of this value depends on what the heat-map represents. In this chapter, there are two different categories of heat-map. First, a *gaze heat-map* $\mathbf{\Gamma}$ is an embedding for head pose information. A value close to one indicates a region of space situated in front of someone's head. Second, an *object heat-map* $\mathbf{\Omega}$ embeds the likelihood for each region of space to contain an object of interest.

***Gaze heat-map* representation $\mathbf{\Gamma}$.** Motivated by the use of cones for modeling the dependency between head pose and gaze [76], we compute a heat-map $\mathbf{\Gamma}_t^n \in [0,1]^{S_U \times S_V}$ for each person $n \in \{1 \dots N_t\}$ by considering a cone whose axis is the direction spanned by the head pan angle ϕ_t^n. Formally, the value of $\mathbf{\Gamma}_t^n$ at any grid cell $\mathbf{p}$ is given by:

$$\mathbf{\Gamma}_t^n(\mathbf{p}) = \begin{cases} 1 & \text{if } |\phi(\mathbf{p}) - \phi_t^n| < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $\phi(\mathbf{p})$ is the angle corresponding to the direction of vector $\overrightarrow{\mathbf{p}_t^n \mathbf{p}}$. The parameter ϵ controls the aperture of the cone. We obtain the total *gaze heat-map* illustrated in Fig. 3.3(b), 3.3(e) and 3.3(h):

$$\mathbf{\Gamma}_t = \frac{1}{N_t} \sum_{n=1}^{N_t} \mathbf{\Gamma}_t^n. \tag{3.3}$$

It is sometimes useful to aggregate the *gaze heat-maps* through time into a mean *gaze heat-map* (see Fig. 3.3(c)) to have a compact representation of the scenario:

$$\mathbf{\Gamma} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{\Gamma}_t. \tag{3.4}$$

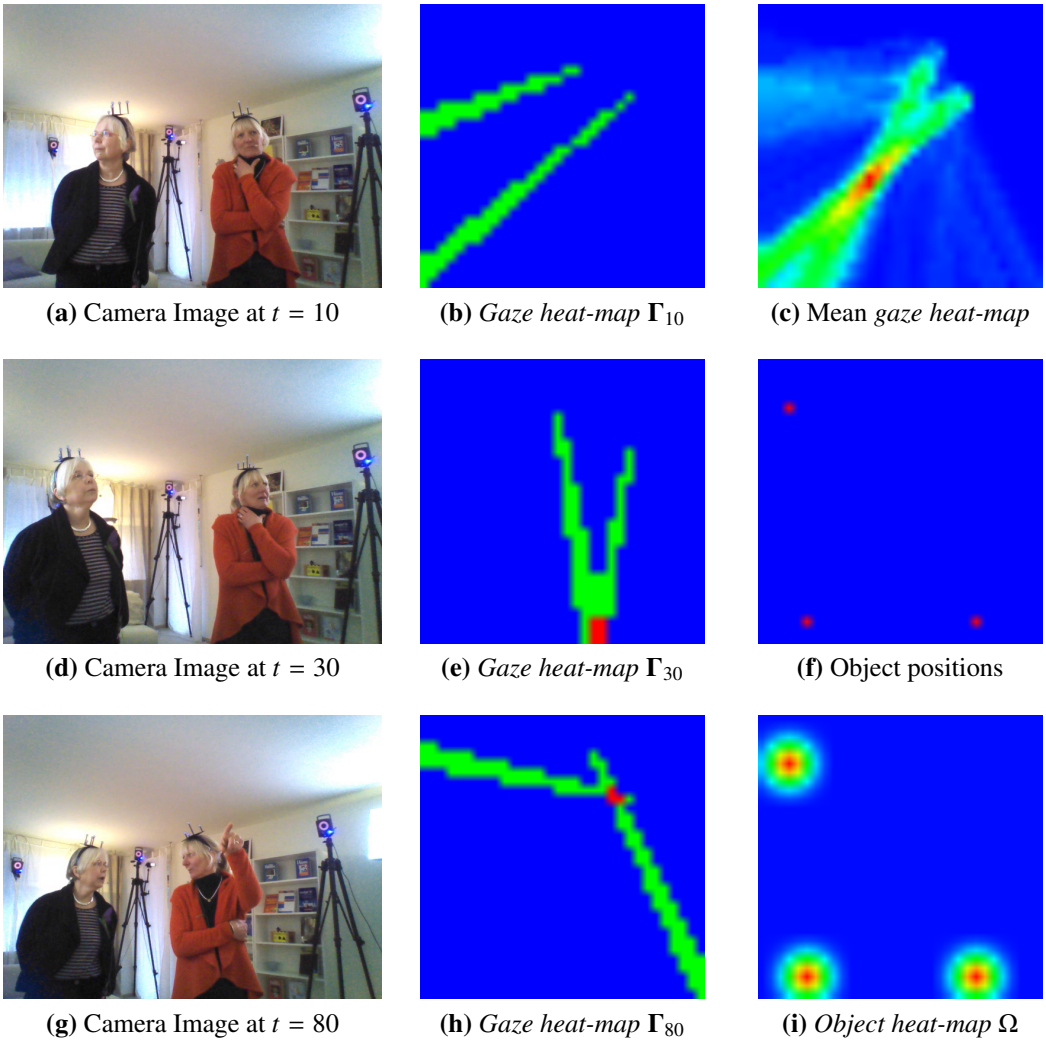
***Object heat-map* $\mathbf{\Omega}$.** Considering a scenario with M objects (*e.g.* Fig. 3.3(f)), we compute a heat-map $\mathbf{\Omega} \in [0,1]^{S_U \times S_V}$ (Fig. 3.3(i)) whose value at grid cell $\mathbf{p}$ is given by:

$$\mathbf{\Omega}(\mathbf{p}) = \max_{1 \le m \le M} \exp\left(-\frac{||\mathbf{p} - \mathbf{p}_{obj}^m||_2^2}{2\sigma_\Omega^2} \right) \tag{3.5}$$

where $\mathbf{p}_{obj}^m$ is the grid cell corresponding to the scene position of the m^{th} object. The variance σ_Ω controls the spread of the peaks. As objects do not move, $\mathbf{\Omega}$ remains constant during a scenario.

Now, let us suppose we have been able to obtain an estimate $\hat{\mathbf{\Omega}}$ of $\mathbf{\Omega}$ from $\mathbf{\Gamma}_1 \dots \mathbf{\Gamma}_T$. Finally, to obtain an actual list of object positions, we extract the local maxima from $\hat{\mathbf{\Omega}}$ and discard local maxima that are too low compared to the global maximum. More precisely, given a candidate position $\mathbf{p}_C$, a neighborhood of this position $\mathcal{N}(\mathbf{p}_C)$ and a shrinking function $\alpha(\cdot)$ such that $\alpha(x) \le x$, we consider that $\mathbf{p}_C$ contains an object if

$$\mathbf{p}_C = \underset{\mathbf{p} \in \mathcal{N}(\mathbf{p}_C)}{\operatorname{argmax}} \hat{\mathbf{\Omega}}(\mathbf{p}) \qquad \text{and} \qquad \hat{\mathbf{\Omega}}(\mathbf{p}_C) \ge \alpha\left(\max_{\mathbf{p}} \hat{\mathbf{\Omega}}(\mathbf{p}) \right). \tag{3.6}$$



(a) Camera Image at $t = 10$ **(b)** *Gaze heat-map* $\mathbf{\Gamma}_{10}$ **(c)** Mean *gaze heat-map*

(d) Camera Image at $t = 30$ **(e)** *Gaze heat-map* $\mathbf{\Gamma}_{30}$ **(f)** Object positions

(g) Camera Image at $t = 80$ **(h)** *Gaze heat-map* $\mathbf{\Gamma}_{80}$ **(i)** *Object heat-map* Ω

Figure 3.3: Illustration of the heat-map representations using a sequence extracted from the *Vernissage* dataset. As can be seen from Fig. 2.3 on page 39, the camera on the Nao robot is located close to the bottom left corner of the *gaze heat-maps*. Heat-map colors range from blue to red to indicate number from 0 to 1. (a), (d), (g) are camera images at different time step from the video. (b), (e), (h) represent the corresponding *gaze heat-maps*. Cone origins in the *gaze heat-maps* indicate people positions; cone axes represent head orientations. (c) is the mean of the *Gaze heat-maps* over the sequence $\frac{1}{T}\sum_{t=1}^{T}\mathbf{\Gamma}_t$. It has been normalized between 0 and 1 to make shades more visible. The object ground truth positions are represented in the heat-map system (f). This provides the ground truth *Object heat-map* (i) used for training and *MSE* evaluation.

The section 3.3.2 below is dedicated to propose a neural network that learns to predict an estimate $\hat{\mathbf{\Omega}}$ of the *object heat-map* from the set of *gaze heat-maps* $\mathbf{\Gamma}_1 \dots \mathbf{\Gamma}_T$.

3.3.2 OBJECT HEAT-MAP INFERENCE

Now, we address the problem of estimating $\hat{\mathbf{\Omega}}$, on which the local maxima detection algorithm can be run. We propose several baselines with justification for their relevance. Then, we present our architectures based on convolutional encoder/decoder.

Heuristics without learning. First, we propose two heuristics with no training. The local maxima detection is performed directly on a combination of the gaze heat-maps. Indeed, the regions that are activated (close to one) in multiple gaze heat-maps are consistently in front of someone's head and have a high chance of containing an object. Previous works [24, 76] already used geometric features based on cone intersections. The heuristics are as follow.

- *Cone*: The local maxima extraction is performed directly on the *mean gaze heat-map* $\mathbf{\Gamma} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{\Gamma}_t$.
- *Intersect*: We define a *gaze intersection heat-map* $\mathbf{\Gamma}_t^{inter}$ per time frame, by setting regions to one only if they are at the intersection of multiple cones. More formally,

$$\mathbf{\Gamma}_t^{inter}(\mathbf{p}) = \begin{cases} 1 & \text{if } \sum_{n=1}^{N_t}\mathbf{\Gamma}_t^n(\mathbf{p}) \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

  The local maxima extraction is performed on $\mathbf{\Gamma}^{inter} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{\Gamma}_t^{inter}$.

Learning-based Baselines. We define some simple regression models. They learn a regression from the *mean gaze heat-map* $\mathbf{\Gamma} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{\Gamma}_t$ to the *Object heat-map* $\hat{\mathbf{\Omega}}$. Both the input and output are considered as vector of $S_U \times S_V$ components.

- *Linear Reg.*: We learn a linear regression model from $\mathbf{\Gamma}$ to $\hat{\mathbf{\Omega}}$. Interestingly, the output of a linear regression is not constrained to lie between 0 and 1, contrary to the definition of $\mathbf{\Omega}$. The local maxima extraction is performed after $\hat{\mathbf{\Omega}}$ has been rescaled in $[0, 1]$.
- *d-FC*: The regression is performed by a network composed of $d \in \{1, 3\}$ fully connected hidden layers of $S_U \times S_V$ units, with ReLU activations. The last hidden layer is fully connected to the output *object heat-map* with sigmoid activations.

Encoder/Decoder Architectures have been used for many computer vision tasks where the goal is to perform a regression between high dimensional spaces [8, 52]. Such architectures are composed of two sub-networks, where the first reduces the spatial resolution of the input to obtain a compact description of it, and the second alternates between

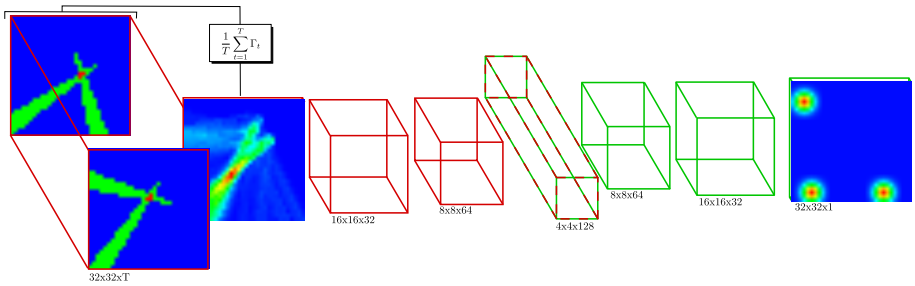
up-sampling and fully-connected layers until recovering a high dimensional output. In our particular problem, we use convolutional layers instead of fully-connected layers to model the spatial connections. Moreover, as the input is a sequence, several encoder architectures can be employed. We propose to use a decoder composed of three successive up-sampling and convolutional layers with 3×3 kernels. The last convolution layer of the decoder employs sigmoid activations. The whole network is trained employing the Mean Squared Error (MSE) loss. We propose the four following architectures that represent a progressively increasing complexity. Graphical representations of the proposed networks are given in Fig. 3.4:

- *Mean-2D-Enc* (Fig. 3.4(a)): This is the simplest model. We use the *mean gaze heat-maps* $\mathbf{\Gamma}$ as in the baselines. It is fed to a standard 2D convolutional encoder composed of three successive convolutional and down-sampling layers.
- *2D-Enc* (Fig. 3.4(b)): In this model, we consider that time plays the role of the color-axis in standard 2D convolutions. $\mathbf{\Gamma}_1 \dots \mathbf{\Gamma}_T$ are concatenated along the third dimension to obtain the *sequence gaze heat-map* $\mathbf{\Gamma}_{1:T}$. Therefore, the first layer kernels have dimension $3 \times 3 \times T$ instead of $3 \times 3 \times 1$ like in *Mean-2D-Enc*.
- *3D-Enc (Fig. 3.4(c))*: Inspired by [55], that shows that 3D convolutions are able to extract reliable features from both the spatial and the temporal dimensions, we propose a 3D-Encoder network on $\mathbf{\Gamma}_{1:T}$. By performing 3D convolutions, the model can capture orientation changes and people motion in successive frames. The time dimension is reduced, from T to 1 after three convolutional and max-pooling layers, before feeding it to the 2D-Decoder.
- *3D/2D U-Net (Fig. 3.4(d))*: This variant of the *3D-Enc* architecture is inspired from the U-Net architecture [102]. In our specific case, since we have a 3D encoder, we need to squeeze the time dimension. To do so, we combine over time the feature maps of the encoder with max-pooling, before concatenation to the decoder.

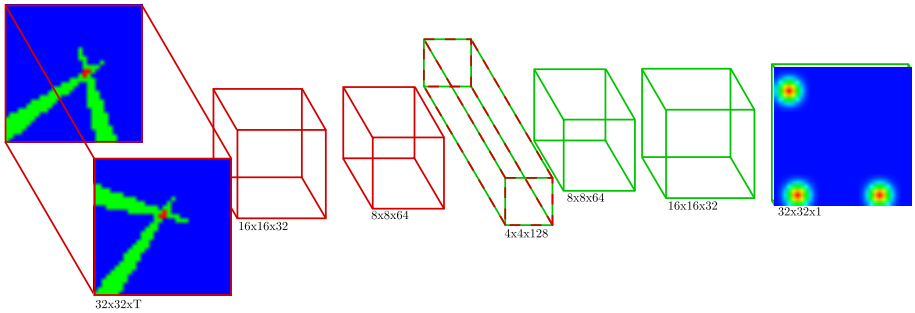
3.4 Synthetic Scenario Generation for Network Training

A large amount of data is required to train deep networks. Unfortunately, obtaining such a dataset is difficult, since, in practice, we would need to know the true object locations for every sequence. For instance, in Vernissage [54], objects outside the field of view have been annotated employing infrared cameras. This setting is well-suited for our problem but it would be difficult to obtain a sufficiently large and diverse dataset of object locations to train deep networks. Consequently, Vernissage is used only to test our model and not to train it.

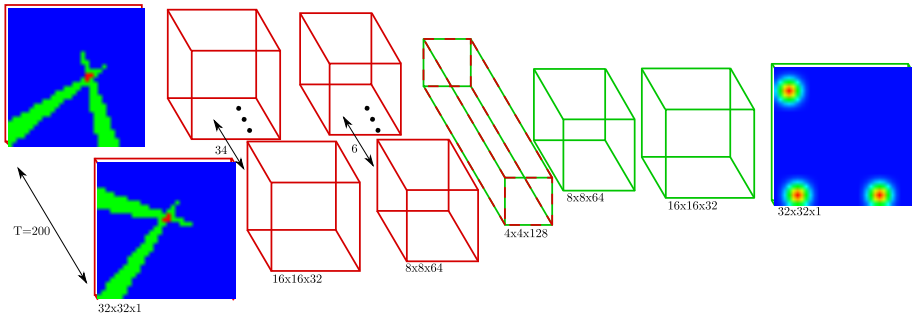
To face this issue, we propose to use synthetically generated data. More precisely, we simulate scenarios involving people and objects, and generate their corresponding input sequences and associated true object locations. We define a probabilistic model that relates the object 2D positions and the head poses, and generate samples according to



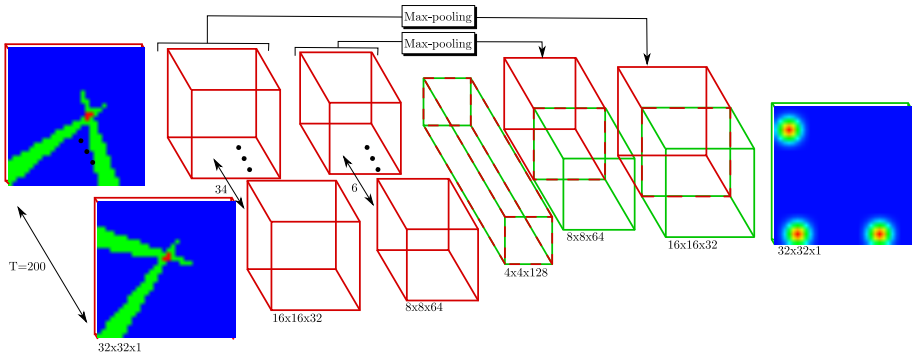
(a) *Mean-2D-Enc*



(b) *2D-Enc*



(c) *3D-Enc*



(d) *3D/2D U-Net*

Figure 3.4: Proposed architectures. More details in section 3.3.2

the underlying distribution. We now aim at generating a scenario of length T involving a constant number N of people with respective positions $\mathbf{x}^n_{1:T}$ and orientations $\phi^n_{1:T}$, given $1 < n < N$; and M objects located at positions $\mathbf{x}^m_{obj}$, $1 < m < M$. To this aim, we define the joint distribution $P(\phi^{1:N}_{1:T}, \mathbf{x}^{1:N}_{1:T}, \mathbf{x}^{1:M}_{obj})$ considering the following factorization:

$$P(\phi^{1:N}_{1:T}, \mathbf{x}^{1:N}_{1:T}, \mathbf{x}^{1:M}_{obj}) = \underbrace{P(\phi^{1:N}_{1:T}|\mathbf{x}^{1:N}_{1:T}, \mathbf{x}^{1:M}_{obj})}_{\substack{\text{Head orientation}\\ \text{distribution}}} \times \underbrace{P(\mathbf{x}^{1:N}_{1:T}|\mathbf{x}^{1:M}_{obj})}_{\substack{\text{People motion}\\ \text{distribution}}} \times \underbrace{P(\mathbf{x}^{1:M}_{obj})}_{\substack{\text{Object position}\\ \text{distribution}}} \tag{3.8}$$

The *object position distribution* $P(\mathbf{x}^{1:M}_{obj})$ is based on a uniform distribution within the top-view grid, since we want to have a high variety of settings. However, some settings are too difficult even for a human to distinguish between objects. For this reason, the generator can choose to resample an object under two criteria. First, the closest two objects are from each other, the highest the chance one of them is resampled. Therefore, we impose that objects have a minimal physical size and that two objects cannot be one above the other. Then, objects too far from the heat-map edges also have a high chance of being resampled. Indeed, in many scenarios, objects of interest tend to be close to the walls, *e.g.* posters, computer screens, paintings in a museum. Moreover, this tends to reduce the number of ambiguous cases in which several objects are aligned from the point of view of someone.

Importantly, in a human-robot interaction scenario, people may look at the robot, but we want to avoid our model to predict the presence of an object at the robot camera position. Therefore, as the camera position $\mathbf{x}_{camera}$ is known, we propose to add a blank object at the corresponding grid cell $\mathbf{p}_{camera}$ in all sequences. The blank object behaves like normal objects – constant position, can be gazed at – but does not appear in the *object heat-map* at training time and thus should be ignored at prediction time. Also, it cannot be resampled while generating the objects.

Concerning the *people motion distribution*, $P(\mathbf{x}^{1:N}_{1:T}|\mathbf{x}^{1:M}_{obj})$, we describe first how the initial positions $\mathbf{x}^{1:N}_1$ are sampled, and then how each $\mathbf{x}^n_{t+1}$ is sampled iteratively from $\mathbf{x}^n_t$. This makes the assumption that people motions are roughly independent. First, the initial positions of people are obtained similarly to object positions. Namely, they are sampled uniformly within the boundaries, and can be resampled when too close to an object, another person, or (contrary to objects) too close to the edges. Concerning the motion, we consider that people can either stay still for a random period of time, or move linearly short distances. In practice, there is a high probability that the person stay still $\mathbf{x}^n_{t+1} = \mathbf{x}^n_t$. Otherwise, $\mathbf{x}^n_{t+\tau}$ is sampled from a normal distribution centered on $\mathbf{x}^n_t$, and possibly resampled as long as $\mathbf{x}^n_{t+\tau}$ is outside the boundaries or too close to another target. In the latter case, $\mathbf{x}^n_{t+1} \dots \mathbf{x}^n_{t+\tau-1}$ are linearly interpolated.

Finally, for the *head orientation distribution*, we propose to adapt the model from chapter 2. Indeed, equations (2.2) to (2.15) represent a generative model that is later transformed with Bayes theorem to achieve an inference algorithm. In the current problem, we know all head and object positions. By setting them all at a constant height, we

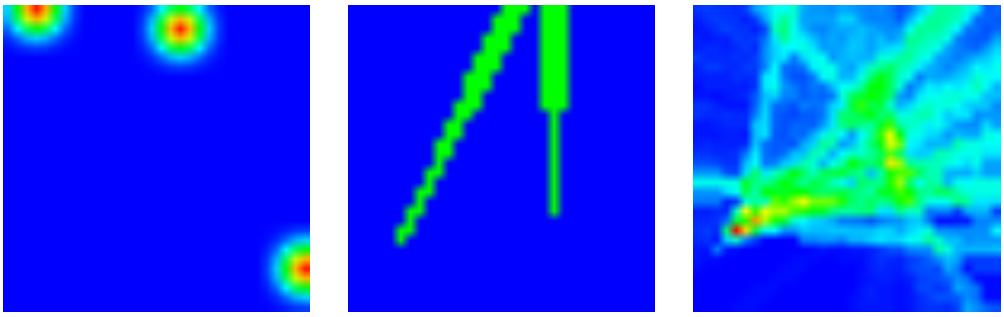
obtain 3D locations $\mathbf{X}_t^n$. Then, for each person n at each time step t, we can define his/her VFOA V_t^n, gaze direction $\mathbf{G}_t^n$ and head reference direction $\mathbf{R}_t^n$. Instead of treating these hidden variables as latent for inference, we choose to sample them. Finally, equation (2.2) gives a distribution that leads to a plausible head orientation sequence $\mathbf{H}_{1:T}^n$ from the sampled gaze directions $\mathbf{G}_{1:T}^n$ and reference directions $\mathbf{R}_{1:T}^n$. In this case, only the head pan angles $\phi_{1:T}^n$ are needed, the tilt angles are discarded (In practice, they are not even computed). Concerning the details of sampling the initial time step, V_1^n is sampled uniformly among the possible targets so that $\mathrm{V}_1^n = j$, then $\mathbf{G}_1^n$ and $\mathbf{R}_1^n$ are set to $\mathbf{X}_1^{nj}$, the direction from person n to target j. For subsequent time steps, the sampling of the hidden variables is based on their respective transition probabilities (2.15), (2.10), (2.11). Finally, if the constraint from (2.29) is violated, $\mathbf{R}_t^n$ and then $\mathbf{H}_t^n$ can be resampled.

Fig. 3.5, 3.6 and 3.7 represent synthetic scenarios, with different setups, generated using this process. In practice, a wide variety of scenarios can be obtained with this approach. For instance, there is no limit to the number of people and/or objects that could be generated in one scenario, except the plausibility of such a scenario with respect to the physical space.

3.5 EXPERIMENTS

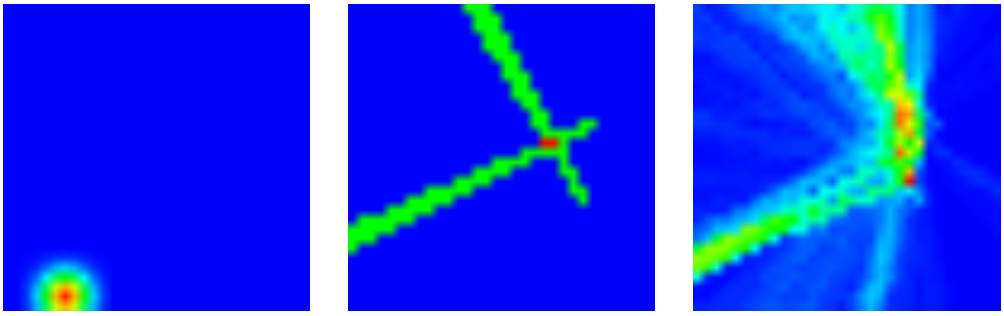
Experiments have been performed both on synthetic data, generated online as described in section 3.4, and on the *Vernissage* dataset [54], described in section 2.6.1 from previous chapter. In particular, for the *Vernissage* dataset, we use head poses either from Vicon data or RGB data. For recall, Vicon data come from calibrated external infrared cameras. RGB data are obtained by detecting faces from the images, estimating head poses in a camera-centered coordinate frame, and projecting back into the scene-centered coordinate frame. Note that, we do not use the datasets employed in [24] and [16] since they are not publicly available.

Implementation details. The heat-map dimensions are set to $S_U = S_V = 32$, to represent a room of size $3m \times 3m$. The cone aperture ϵ is set to $2°$. We fixed the input sequence size to $T = 200$ time steps. On the *Vernissage* dataset, the videos are subsampled to 5 fps, then the duration of a sequence is $40s$ and we can extract several sequences from each video sequence. By using a sliding window and 50% overlap, we extract a total of 224 sequences. We use the visual focus of attention annotations to obtain the true objects of interest for each sequence. Consequently, the number of objects can vary from 1 to 3 in the test sequences. We employ the adam optimizer [57] for 10 epochs. For all neural network architectures employed in the experiments, the batch size is set to 32. In all cases, we perform the exact same local maxima extraction method as in (3.6) after estimating $\hat{\mathbf{\Omega}}$ to obtain the list of object positions. The neighborhood $\mathcal{N}(\cdot)$ in eq. (3.6) is defined as a sliding region of 5×5 pixels, and the shrinking function $\alpha : x \mapsto \ln(1 + x)$. In all our experiments, we report *Precision* and *Recall*, and these two metrics are combined to obtain the *f1-score*. *Precision* measures the percentage of detected objects that are true objects.



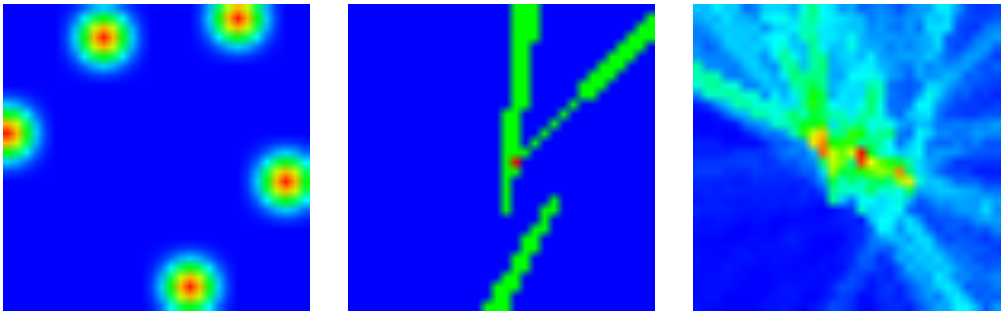
(a) *Object heat-map* **(b)** *Gaze heat-map* **(c)** Mean *gaze heat-map*

Figure 3.5: Heat-maps from a synthetic scenario generated randomly, with 2 people ($N = 2$) and 3 objects ($M = 3$). (a): the ground truth *Object heat-map* $\mathbf{\Omega}$ used for training or evaluation. (b): a *Gaze heat-map* randomly chosen among the sequence. (c): the mean *gaze heat-map* over the sequence.



(a) *Object head-map* **(b)** *Gaze heat-map* **(c)** Mean *gaze heat-map*

Figure 3.6: Heat-maps from a synthetic scenario generated randomly, similar to Fig. 3.5, but with a different setup: 2 people ($N = 2$) and 1 object ($M = 1$).



(a) *Object head-map* **(b)** *Gaze heat-map* **(c)** Mean *gaze heat-map*

Figure 3.7: Heat-maps from a synthetic scenario generated randomly, similar to Fig. 3.5, but with a different setup: 3 people ($N = 3$) and 5 objects ($M = 5$).

Recall measures the percentage of true objects correctly detected. In order to compute these metrics, we employ a Hungarian algorithm that matches the detections with the real objects positions based on their respective distances. Importantly, the detection is considered as a success if the distance between the estimated and annotated distances is lower than 50cm in the real-world space. For all learning-based approaches, we also report the MSE between the predicted and true *object heat-maps*.

Results and Discussion. In Table 3.1, we report the results obtained employing all methods described on both *synthetic* and real data.

It has to be noted that many different recurrent architectures have been considered, either alone or in conjunction with one of the proposed convolutional Encoder/Decoder architectures *e.g*. adapted from the convolutional LSTM [30]. All of them converged to networks predicting always the same (or almost the same) *object heat-map*. We believe that, in this formulation, the ability to combine information from distant time frame is important, and this is difficult to achieve with RNN (or LSTM) processing data sequentially [91].

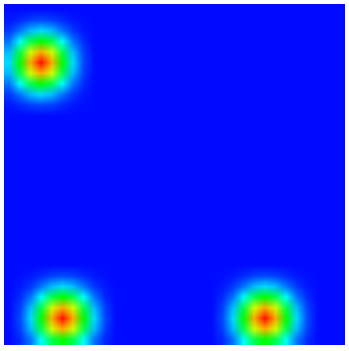
First, we notice that using computer vision head pose estimator or the more accurate Vicon data changes only marginally the final result. There is even a small drop in performance when using accurate head poses, mostly on methods that are insensitive to time order. In this cases, the noise from head pose estimations may randomly add some directions corresponding to an undetected object. Anyway, since both results are very similar, we conclude that the method is robust to slight noise in head pose estimation. We now indistinctly refer to *Vernissage* results for experiments on *Vernissage* dataset using either RGB or Vicon data.

From the experiments, we observe that learning-based approaches clearly outperform those based on cone intersections inspired from [24]. Indeed, even on the synthetic datasets, their *precision* and *recall* do not reach better than 18.8% and 53.9% respectively, whereas a simple linear regression reaches considerably higher scores (50.5% and 76.9% respectively). The same remark stands for the *Vernissage* dataset. Increasing the network complexity by simply adding fully-connected layers does not bring any improvement and even reduce the performance. Then, we observe that all proposed encoder/decoder models clearly outperform other methods by a substantial margin on the synthetic dataset. There, we obtain a 22.3% gain in terms of f1-score when employing the *3D/2D U-Net* with respect to the linear regression model. On the *Vernissage* dataset, a 5.1% gain is obtained in terms of *f1-score* when employing the *Mean-2D-Enc* with respect to the linear regression model. These experiments validate the use of the encoder/decoder architecture.

We notice that the performance on the synthetic dataset increases with encoder complexity. However, the inverse phenomenon is observed on *Vernissage*, where the best performances are obtained using the simplest encoder architecture (that does not model time). Our guess for this observation is that there is a significant discrepancy between the distribution of *Vernissage* data and the *synthetic* data distribution sampled according to (3.8). Therefore, more complex models probably tend to over-fit the *synthetic* data

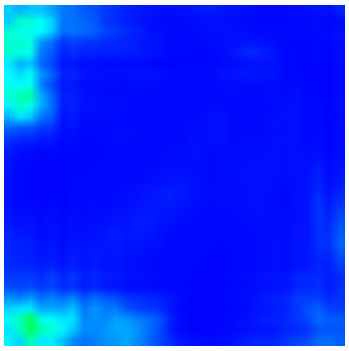
Dataset	Synthetic			
Method	*MSE*	*Precision*	*Recall*	*f1-score*
Cone	-	18.8	53.9	27.8
Intersect	-	21.1	35.0	26.3
Linear Reg.	1.25 ± 0.02	50.5 ± 2.2	76.9 ± 1.0	60.9 ± 1.8
1-FC	1.06 ± 0.03	64.9 ± 1.6	61.5 ± 1.5	63.1 ± 1.1
3-FC	1.05 ± 0.01	65.9 ± 0.6	59.9 ± 2.2	62.8 ± 1.2
Mean-2D-Enc	1.00 ± 0.03	74.5 ± 2.4	59.5 ± 1.7	66.1 ± 1.3
2D-Enc	0.98 ± 0.02	76.8 ± 2.2	62.2 ± 1.5	68.7 ± 1.7
3D-Enc	0.85 ± 0.06	88.2 ± 3.9	71.4 ± 2.1	78.9 ± 2.4
3D/2D U-Net	**0.75 ± 0.01**	**89.0 ± 1.2**	**78.0 ± 0.6**	**83.2 ± 0.8**
Dataset	Vernissage Vicon data			
Method	*MSE*	*Precision*	*Recall*	*f1-score*
Cone	-	16.7	34.8	22.5
Intersect	-	17.1	17.6	17.3
Linear Reg.	1.49 ± 0.03	36.4 ± 3.7	**51.9 ± 2.6**	42.8 ± 3.4
1-FC	1.50 ± 0.02	33.4 ± 1.6	35.3 ± 2.3	34.3 ± 1.7
3-FC	1.49 ± 0.03	30.0 ± 3.5	30.1 ± 1.6	30.0 ± 2.5
Mean-2D-Enc	**1.39 ± 0.03**	**54.8 ± 1.4**	39.7 ± 1.8	**46.0 ± 1.6**
2D-Enc	1.42 ± 0.03	50.0 ± 5.0	38.7 ± 3.2	43.6 ± 3.8
3D-Enc	1.44 ± 0.03	51.2 ± 3.3	38.5 ± 3.5	43.9 ± 3.4
3D/2D U-Net	1.47 ± 0.04	47.1 ± 4.4	41.3 ± 1.5	44.0 ± 2.6
Dataset	Vernissage RGB data			
Method	*MSE*	*Precision*	*Recall*	*f1-score*
Cone	-	20.7	35.8	26.2
Intersect	-	34.9	27.2	30.6
Linear Reg.	1.48 ± 0.04	37.0 ± 4.9	**53.7 ± 5.0**	43.7 ± 4.6
1-FC	1.49 ± 0.02	29.9 ± 3.2	35.2 ± 2.5	32.3 ± 2.8
3-FC	1.49 ± 0.02	28.0 ± 3.5	29.9 ± 1.5	28.8 ± 2.4
Mean-2D-Enc	**1.37 ± 0.02**	**60.1 ± 1.5**	41.1 ± 1.0	**48.8 ± 1.2**
2D-Enc	1.39 ± 0.03	54.9 ± 4.2	40.5 ± 1.6	46.6 ± 2.5
3D-Enc	1.43 ± 0.05	49.9 ± 8.1	37.1 ± 9.0	42.5 ± 8.7
3D/2D U-Net	1.44 ± 0.04	45.1 ± 4.8	38.5 ± 2.2	41.5 ± 3.3
Dataset	Brau et al. [16]			
Method	*MSE*	*Precision*	*Recall*	*f1-score*
Brau et al. [16]	-	59.0	48.0	*52.9*

Table 3.1: Results obtained on the proposed synthetically generated dataset and on the *Vernissage* dataset [54]. MSE values reported were multiplied by 10^2 to facilitate reading. *Precision*, *recall* and *f1-score* represent percentages. For learning-based approaches, we report the mean and standard deviation over five runs. Results from [16] on their own dataset are reported for comparison.

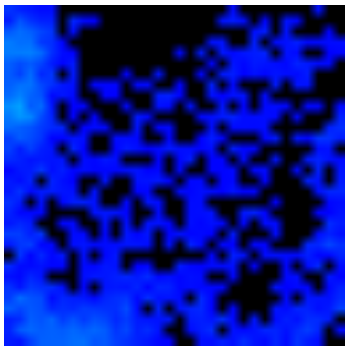


(a) *Object heat-map* $\mathbf{\Omega}$

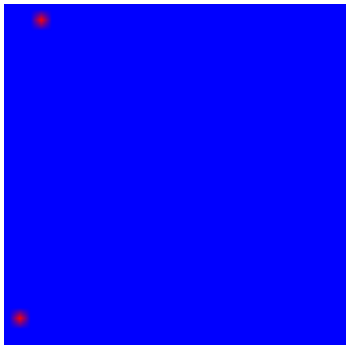
(b) $\hat{\mathbf{\Omega}}$ - *Mean-2D-Enc*



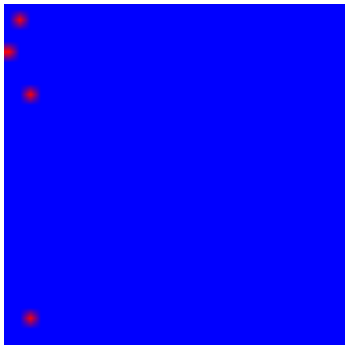
(c) $\hat{\mathbf{\Omega}}$ - *3D/2D U-Net*



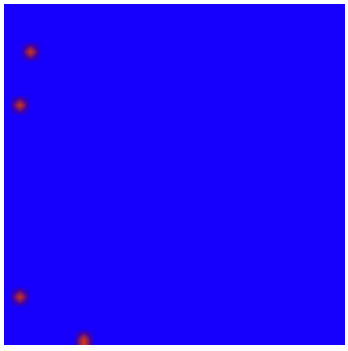
(d) $\hat{\mathbf{\Omega}}$ - *Linear Reg.*



(e) Objects - *Mean-2D-Enc*

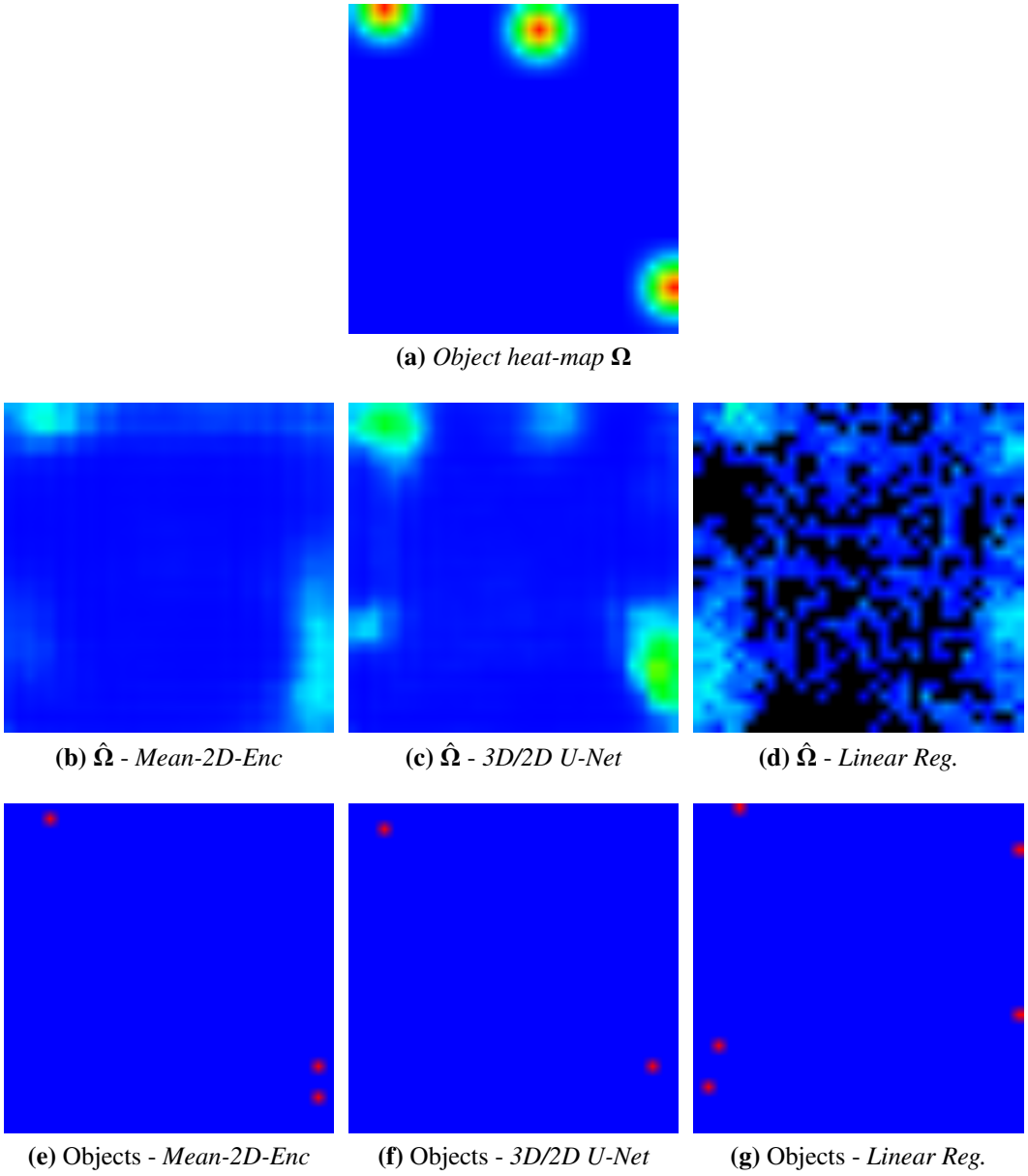


(f) Objects - *3D/2D U-Net*



(g) Objects - *Linear Reg.*

Figure 3.8: Application of three methods on the sequence from *Vernissage* dataset, illustrated in Fig. 3.3. The *object heat-map* (a) is duplicated from Fig. 3.3(i) for better readability. (b), (c), (d): Estimates of the *object heat-map* $\hat{\mathbf{\Omega}}$ using three different architectures. (e), (f), (g): Corresponding objects positions, obtained as the highest local maxima from $\hat{\mathbf{\Omega}}$. Black pixels in (d) indicate negative values.



(a) *Object heat-map* $\mathbf{\Omega}$

(b) $\hat{\mathbf{\Omega}}$ - *Mean-2D-Enc* **(c)** $\hat{\mathbf{\Omega}}$ - *3D/2D U-Net* **(d)** $\hat{\mathbf{\Omega}}$ - *Linear Reg.*

(e) Objects - *Mean-2D-Enc* **(f)** Objects - *3D/2D U-Net* **(g)** Objects - *Linear Reg.*

Figure 3.9: Application of three methods on the synthetic sequence illustrated in Fig. 3.5. The *object heat-map* (a) is duplicated from Fig. 3.5(a) for better readability. (b), (c), (d): Estimates of the *object heat-map* $\hat{\mathbf{\Omega}}$ using three different architectures. (e), (f), (g): Corresponding objects positions, obtained as the highest local maxima from $\hat{\mathbf{\Omega}}$. Black pixels in (d) indicate negative values.

distribution, and thus transfer less well on the *Vernissage* dataset. More realistic training data could lead to further improvements. This could be obtained by gathering a dataset of real-life scenarios which could be use either as training data or to improve the quality of the generative model.

The only methods from the literature that we are aware of are [24] and [16]. In both cases, neither the data nor the code have been made available online. Moreover, the papers lack information about parameters or hyperparameters that prevented us to test it. Additionally, [16] explicitly discarded the *Vernissage* dataset in their experiments. Results on their dataset (59% precision and 48% recall) are comparable to ours on *Vernissage*. Note that, [16] employed a larger success threshold (1.0m in the real-world space for 50cm in our case) and consequently would obtain lower scores according to our evaluation protocol. We wish to test our method on their dataset in the future. We do not compare to [24] since they did not report any quantitative results on location estimation.

In Fig. 3.8, the predicted *gaze heat-maps* $\hat{\mathbf{\Omega}}$ for several learning-based approaches applied on the scenario from Fig. 3.3 are displayed. The architectures *Mean-2D-Enc* and *Linear Reg.* use the average *gaze heat-map* $\frac{1}{T}\sum_{t=1}^{T}\mathbf{\Gamma}_t$ as input, whereas *3D/2D U-Net* takes the whole concatenated sequence $\mathbf{\Gamma}_{1:T}$. All three approaches are approximately able to predict the positions of two objects of interest. The third object is probably not targeted enough during the sequence to be found. The black pixels in the *Linear Regression* indicate negative values. All other approaches end with a sigmoid activation so each pixel value is homogeneous to a probability. The lower number of falsely proposed object positions for the *Mean-2D-Enc* is consistent with the higher mean precision reported. Comparatively, we show in Fig. 3.9 the same approaches, with the same training weights, applied on the synthetic scenario from Fig. 3.5. We observe in this case that the *3D/2D U-Net* yields an *object heat-map* $\hat{\mathbf{\Omega}}$ closer to the expected one $\mathbf{\Omega}$ than the other models, and lead to a higher precision.

We also report experiments to measure the impact in performance of the sequence length T in Fig 3.10. Precisely, we selected *Mean-2D-Enc* (as best model on *Vernissage*) and *3D/2D U-Net* (as best model on *synthetic*) and compute the *f1-score* evolution for these two networks varying T from 10 to 450. Both networks behave similarly to the results reported before: *3D/2D U-Net* is consistently better on *synthetic* data than *Mean-2D-Enc*, and consistently worse on the *Vernissage* dataset. We observe that the performances of both networks tend to increase with the sequence length on *synthetic* data, though quite slowly for $T > 150$. However, when the networks are transfered to be used on the *Vernissage* dataset, the *f1-score* stops increasing past $T = 200$ or 250. Moreover, the variances are sometimes quite higher, which could indicate a more unstable training process. This validates the choice of $T = 200$ for our experiments.

3.6 Conclusions

We defined the problem of *unconstrained gaze-following* as finding the locations of objects of interest solely from gaze direction of visible people. Importantly, this allows

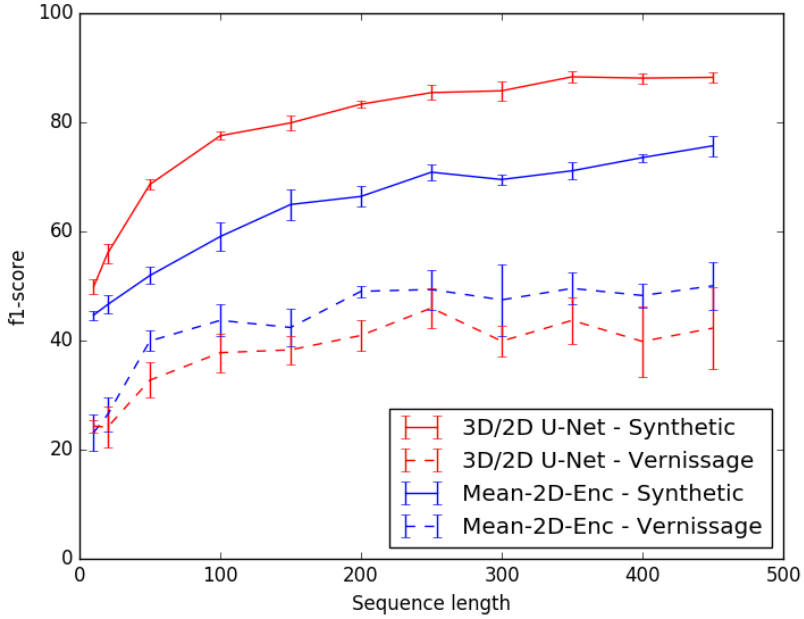


Figure 3.10: Performance obtained on the *synthetic* and *Vernissage* datasets with RGB data. We measure the *f1-score* with different values of sequence length T.

for finding objects outside the field-of-view. In this context, we propose a novel spatial representation for head poses (approximating gaze direction) and object locations. This representation is based on probability heat-maps, and uses a top-view perspective instead of the image plane as in most previous works. We have presented a framework that takes advantage of convolutional encoder/decoder architectures to learn the spatial relationship between head poses and object locations. We compare nine different methods on synthetic and real data and conclude that learning-based approaches outperform geometry-based ones. We also demonstrate that the necessary training examples can be quickly and easily obtained through a synthetic data generation process.

We believe this work will open new perspectives for research. In particular, several decisions were taken to obtain an end-to-end method (*e.g.* heat-map representation or elevation coordinate omission), which makes it hardly suitable in some situations. The unconstrained gaze-following problem would benefit greatly from a benchmark of different representations and inference models, and of the influence of each simplifying hypothesis. In parallel, the availability of suitable datasets would ease future research on this topic.

CHAPTER 4

DEEP REINFORCEMENT LEARNING FOR AUDIO-VISUAL ROBOT GAZE CONTROL IN HUMAN-ROBOT INTERACTION

4.1 INTRODUCTION

Human-robot interaction (HRI) is a growing field that combines perception and action for the robotic agent. In the previous chapter, we described a method to determine where objects of interest are situated solely from gaze direction of people. However, it is assumed people are visible most of the time, so that the head pose can be tracked. In an actual interaction, people may not be nor stay in front of the robot. For instance, when the robot plays the role of an art guide, as simulated in the *Vernissage* dataset, visitors may approach from different paths. People are sometimes too far from each other for the robot to see everyone at once. More complicated scenarios, *e.g.* having the role of a waiter, require the robot to move and follow some complicated strategy to maximize information along the way.

Until now, we have mainly focused on visual data. Let's recall that the robot also possesses other sensors. In particular, robots are generally equipped with microphones and proprioceptive sensors. Audio information is specially interesting since it is complementary with visual data. For instance, using multiple microphones, it is possible to infer the origin of a specific sound, like someone speaking, even when the sound source lies outside of the camera field of view or is occluded. On the other hand, audio information is sparse and subject to intrinsic noise from reverberation. Proprioceptive sensors, also called self-motion sensors, provide the motor state of robotic joints. This helps the robot correct for its own displacement while analyzing other kind of data. Since people being outside of the field of view is often a problem, it is promising to combine data from other sensors to help vision accuracy. Nevertheless, balancing the role of each sensor is not an easy task, for instance when they provide contradictory information, and can hardly be handcrafted. Indeed, handling all the possible situations with a set of handcrafted rules

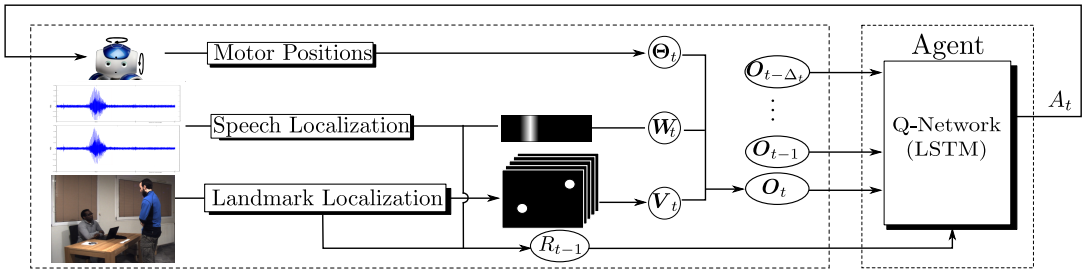


Figure 4.1: Overview of the proposed deep RL method for controlling the gaze of a robot. At each time index t, audio and visual data are represented as binary maps which, together with motor joint angles, form the set of observations $\boldsymbol{O}_t$. A motor action A_t (rotate the head left, right, up, down, or stay still) is selected based on past and present observations via maximization of current and future rewards. The rewards R are based on the number of visible persons as well as on the presence of speech sources in the camera field of view. We use a deep Q-network (DQN) model that can be learned both off-line and on-line. Please refer to Sections 4.3.1 and 4.3.2 for the mathematical notations and detailed problem formulation.

would be laborious and most-likely sub-optimal. Moreover, it is very difficult to predict how the decision process will change *e.g.* with the illumination or the acoustics of the room.

Gaze Control represents the set of strategies in which a person moves the head and the eyes to shift his/her gaze [46]. Applied to robotics, it denotes the way the robot turn its visual sensor along time. It makes the robot actively decide where to look at, therefore controlling how it *perceives* the environment. To do so, the robot moves its own motor joints and so *modifies* the environment. Actually, it is known that the robot's gaze behavior has a strong effect on the turn-taking conduct of the participants [115] and can be used to communicate.

In this chapter, we address the problem of robotic gaze control for social HRI. In particular, we want the robot to control the orientation of its head (and the associated camera) to behave properly during informal group gatherings. We propose a methodology for the robot to autonomously learn strategies that lead to focusing group of people using audio-visual information. More specifically, we want a robot to learn to find people in the environment, hence maximize the number of people present in its field of view, and favor people who speak. We believe this could be useful in many real scenarios, such as a conversation between a companion robot and a group of persons, where the robot needs to learn to look at people, in order to behave properly. The reason for using multiple sources of information can be found in recent HRI research suggesting that no single sensor can reliably serve robust interaction [94]. Importantly, when it comes to the employment of several sensing modalities in complex social interactions, it becomes difficult to implement an optimal policy based on handcrafted rules that take into consideration all possible situations that may occur. On the contrary, we propose to follow a data-driven approach to face such complexity. We address this problem using a *reinforcement learning* (RL) approach [120]. RL is a machine learning paradigm in which agents learn by themselves by trial-and-error to achieve successful strategies. As opposed to supervised learning, there is no need for optimal decisions at training time, only a *reward*, *i.e.* a

unidimensional value that evaluates how good a decision is. This paradigm, inspired from behavioral psychology, may enable a robot to autonomously learn a policy that maximizes accumulated rewards. In our case, the agent, a robot companion, autonomously moves its head depending on its knowledge about the environment. This knowledge is called the *agent state*, and it is defined as a sequence of audio-visual observations, motor readings, actions, and rewards. In practice the optimal policy for making decisions is learned from the reward computed using detected faces of participants and sound sources being localized. The use of annotated data is not required to learn the best policy as the agent learns autonomously by trial-and-error in an unsupervised manner. Moreover, using our approach, it is not necessary to make any assumption about the number of people as well as their locations in the environment.

The use of RL techniques presents several advantages. First, training using optimal decisions is not required since the model learns from the reward obtained from previous decisions. The reward may well be viewed as a feedback signal that indicates how well the robot is doing at a given time step. Second, the robot must continuously make judgments so as to select good actions over bad ones. In this sense, the model can keep training at test time and hence it benefits from a higher adaptation ability. Finally, we avoid the need to resort to an annotated training set or calibration data. In our opinion, it seems entirely natural to use RL techniques to "educate" a robot, since recent neuroscientific studies have suggested that reinforcement affects the way infants interact with their environment, including what they look at [3], and that gazing at faces is not innate, but that environmental importance influences the gazing behavior.

The contributions of this chapter are the followings. First, robot gaze control is formulated as a reinforcement learning problem, allowing the robot to autonomously learn its own gaze control strategy from multimodal data. Second, we use *deep* reinforcement learning to model the action-value function, and suggest several architectures based on a recurrent neural network model called Long Short-Term Memory (LSTM) that allow us to experiment with both early- and late-fusion of audio and visual data. Third, we introduce a simulated environment that enables us to learn the proposed deep RL model without the need of repeatedly spending hours of tedious interaction. Finally, by experimenting with both a publicly available dataset and with a real robot, we provide empirical evidence that our method achieves state-of-the-art performance.

4.2 Related Work

The concept of gaze control has its roots in active vision [2], or more generally in active perception [10]. These research fields are interested in how an agent (human [36], robot, animal, etc.) use its sensing feedback to take actions that will enhance its perception of the environment. Many robotic applications exist [21]. Gaze control is closely related to active vision, but is preferred when perception is not the only matter. For instance, in a social situation, moving the head can have several interpretations besides perception augmentation. Robotic gaze control has been addressed in the framework of sensor-based

servoing. In [13] an ad-hoc algorithm is proposed to detect, track, and involve multiple persons into an interaction, combining audio-visual observations. In a multi-person scenario, [12] investigated the complementary nature of tracking and visual servoing that enables the system to track several persons and to visually control the gaze to keep a selected person in the camera field of view. Also, in [136], a system for gaze control of socially interactive robots in multiple-person scenarios is presented. This method requires external sensors to locate human participants.

Reinforcement Learning has been successfully employed in different domains, including robotics [58]. The RL goal is to find a function, called a policy, which specifies which action to take in each state, so as to maximize some function (*e.g.*, the mean or expected discounted sum) of the sequence of rewards. Therefore, learning the suitable policy is the main challenge, and there are two main categories of methods to address it. First, policy-based methods define a space from the set of policies, and sample policies from this space. The reward is then used, together with optimization techniques, *e.g.* gradient-based methods, to increase the quality of subsequent sampled policies [130]. Second, value-based methods consist in estimating the expected reward for the set of possible actions, and the actual policy uses this value function to decide the suitable action, *e.g.* choose the action that maximizes the value-function. In particular, popular value-based methods include Q-learning [128] and its deep learning extension, Deep Q-Networks (or DQNs) [81].

There are several RL-based HRI methods relevant to our work. In [41] an RL algorithm is used for a robot to learn to play a game with a human partner. The algorithm uses vision and force/torque feedback to choose the motor commands. The uncertainty associated with human actions is modeled via a Gaussian process model, and Bayesian optimization selects an optimal action at each time step. In [80] RL is employed to adjust motion speed, timing, interaction distances, and gaze in the context of HRI. The reward is based on the amount of movement of the subject and the time spent gazing at the robot in one interaction. As external cameras are required, this cannot be easily applied in scenarios where the robot has to keep learning in a real environment. Moreover, the method is limited to the case of a single human participant. Another example of RL applied to HRI can be found in [122], where a human-provided reward is used to teach a robot. This idea of interactive RL is also exploited in [28] in the context of a table-cleaning robot. Visual and speech recognition are used to get advice from a parent-like trainer to enable the robot to learn a good policy efficiently. An explicit reward is used in [104] to learn how to point a camera towards the active speaker in a conversation. Audio information is used to determine where to point the camera, while the reward is provided using visual information: the active speaker raises a blue card that can be easily identified by the robot. The use of a multimodal deep Q-network (DQN) to learn human-like interactions is proposed in both [96] and [97]. The robot must choose an action to shake hands with a person. The reward is either negative, if the robot tries unsuccessfully to shake hands, positive, if the hand-shake is successful, or null otherwise. In practice, the reward is obtained from a sensor located in the hand of the robot and it takes fourteen training days to learn this skill successfully. Finally in [125], the authors use an RL approach to learn good policies to control the orientation of a mobile robot during social group

conversations. The robot learns to turn its head towards the speaking person. However, their model is learned on simulated data that are restricted to a few predefined scenarios with static people and a predefined spatial organization of the groups.

In contrast to all these works, we aim at learning an optimal gaze control behavior using minimal supervision provided by a reward function, instead of adopting a handcrafted gaze control strategy. Importantly, our model requires neither external sensors nor human intervention to compute the reward, allowing the robot to autonomously learn where to gaze.

4.3 Reinforcement Learning for Gaze Control

4.3.1 Problem Formulation

We consider a robot whose goal is to have a behavior that maximizes the social information extracted from its sensors. In this work, we simplify the problem with two simplifying hypotheses. First, most social information comes from people faces; second, all people are important, in particular speaking ones. In this case, the goal becomes looking at a group of people. Hence, the robot must learn by itself a gazing strategy via trials and errors. The desired robot action is to rotate its head (endowed with a camera and four microphones) to maximize the number of persons lying in the camera field-of-view. Moreover, the robot should prefer to look at speaking people instead of silent ones. The terms *agent* and *robot* will be used indistinctly.

Random variables and their realizations are denoted with uppercase and lowercase letters, respectively. Vectors and matrices are in bold italic. At each time index t, the agent gathers motor joint $\boldsymbol{\Theta}_t$, visual $\boldsymbol{V}_t$, and audio $\boldsymbol{W}_t$ observations and performs an action $A_t \in \mathcal{A}$ from an action set according to a policy π, i.e. controlling the two head motors such that the robot gazes in a selected direction. Once an action is performed, the agent receives a reward R_t, as explained in detail below.

Without loss of generality we consider the companion robot Nao whose head has two rotational degrees of freedom: motor readings correspond to pan and tilt angles, $\boldsymbol{\Theta}_t = (\phi_t, \theta_t)$. The values of these angles are relative to a reference head orientation, e.g. aligned with the robot body. This reference orientation together with the motor limits define the robot-centered *motor field-of-view* (M-FOV).

We use the multiple person detector of [19] to estimate two-dimensional visual landmarks, i.e. image coordinates, for each detected person, namely the nose, eyes, ears, neck, shoulders, elbows, wrists, hip, knees and ankles, or a total of $J = 18$ possible landmarks for each person. Based on the detection of these landmarks, one can determine the number of (totally or partially) observed persons, N_t, as well as the number of observed faces, F_t. Note that in general the number of faces that are present in the image (i.e. detection of nose, eyes or ears) may be smaller than the number of detected persons. Since the camera is mounted onto the robot head, the landmarks are described in a head-centered reference system. Moreover, these landmarks are represented by J binary maps of size $K_v \times L_v$,

namely $\boldsymbol{V}_t \in \{0,1\}^{K_v \times L_v \times J}$, where 1 (resp. zero) corresponds to the presence (resp. absence) of a landmark. Notice that this representation gathers all the detected landmarks associated with the N_t detected persons.

Audio observations are provided by the multi audio-source localization method described in [69]. Audio observations are also represented with a binary map of size $K_a \times L_a$, namely $\boldsymbol{W}_t \in \{0,1\}^{K_a \times L_a}$. A map cell is set to 1 if a speech source is detected at that cell and 0 otherwise. The audio map is robot-centered and hence it remains fixed whenever the robot turns its head. Moreover, the audio map spans an *acoustic field-of-view* (A-FOV), which is much wider than the *visual field-of-view* (V-FOV), associated with the camera mounted onto the head. The motor readings allow us to estimate the relative alignment between the audio and visual maps and to determine whether a speech source lies within the visual field-of-view or not. This is represented by the binary variable $\Sigma_t \in \{0,1\}$, such that $\Sigma_t = 1$ if a speech source lies in the visual field-of-view and $\Sigma_t = 0$ if none of the speech sources lies inside the visual field-of-view.

Let $\boldsymbol{O}_t = \{\boldsymbol{\Theta}_t, \boldsymbol{V}_t, \boldsymbol{W}_t\}$ and let $\boldsymbol{S}_t = \{\boldsymbol{O}_1, \ldots, \boldsymbol{O}_t\}$ denote the state variable. Let the set of actions be defined by $\mathcal{A} = \{\varnothing, \leftarrow, \uparrow, \rightarrow, \downarrow\}$, namely either remain in the same position or turn the head by a fixed angle in one of the four cardinal directions. We propose to define the reward R_t as follows:

$$R_t = F_{t+1} + \alpha \Sigma_{t+1}, \tag{4.1}$$

where $\alpha \geq 0$ is an adjustment parameter. Large α values return high rewards when speech sources lie within the camera field-of-view. We consider two types of rewards which are referred to in Section 4.4 as *Face_reward* ($\alpha = 0$) and *Speaker_reward* ($\alpha = 1$). Notice that the number of observed faces, F_t, is independent of the speaking state of each person. Upon the application at hand, the value of α allows one to weight the importance given to speaking persons.

In RL, the model parameters are learned on sequences of states, actions and rewards, called episodes. At each time index t, an optimal action A_t should be chosen by maximizing the immediate and future rewards, $R_t, R_{t+1}, \ldots, R_T$. We make the standard assumption that future rewards are discounted by a factor γ that defines the importance of short-term rewards as opposed to long-term ones. We define the discounted future return $\bar{R}_t$ as the discounted sum of future rewards, $\bar{R}_t = \sum_{\tau=0}^{T-t} \gamma^\tau R_{\tau+t}$. If $\gamma = 0$, $\bar{R}_t = R_t$ and, consequently, we aim at maximizing only the immediate reward whereas when $\gamma \approx 1$, we favor policies that lead to better rewards in the long term. Considering a fixed value of γ, we now aim at maximizing $\bar{R}_t$ at each time index t. In other words, the goal is to learn a policy, $\pi(a_t, \boldsymbol{s}_t) = P(A_t = a_t | \boldsymbol{S}_t = \boldsymbol{s}_t)$ with $(a_t, \boldsymbol{s}_t) \in \mathcal{A} \times \mathcal{S}$, such that if the agent chooses its actions according to the policy π, the expected $\bar{R}_t$ should be maximized. The Q-function (or the action-value function) is defined as the expected future return from state $\boldsymbol{S}_t$, taking action A_t and then following any given policy π:

$$Q_\pi(\boldsymbol{s}_t, a_t) = \mathbb{E}_\pi[\bar{R}_t | \boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t]. \tag{4.2}$$

Learning the best policy corresponds to the following optimization problem $Q^*(\boldsymbol{s}_t, a_t) = \max_\pi [Q_\pi(\boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t)]$. The optimal Q-function obeys the identity known as the

Bellman equation:

$$Q^*(\boldsymbol{s}_t, a_t) = \mathbb{E}_{S_{t+1}, R_t}\Big[R_t + \gamma \max_a (Q^*(\boldsymbol{S}_{t+1}, a))\Big|\boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t\Big]. \tag{4.3}$$

This equation corresponds to the following intuition: if we have an estimator $Q^*(\boldsymbol{s}_t, a_t)$ for $\bar{R}_t$, the optimal action a_t is the one that leads to the largest expected $\bar{R}_t$. The recursive application of this policy leads to equation (4.3). A straightforward approach would consist in updating Q at each training step i with:

$$Q^i(\boldsymbol{s}_t, a_t) = \mathbb{E}_{S_{t+1}, R_t}\Big[R_t + \gamma \max_a (Q^{i-1}(\boldsymbol{S}_{t+1}, a))|\boldsymbol{S}_t = \boldsymbol{s}_t, A_t = a_t\Big]. \tag{4.4}$$

Following equation (4.4), we estimate each action-value $Q^i(\boldsymbol{s}_t, a_t)$ given that we follow, for the next time steps, the policy implied by Q^{i-1}. In practice, we approximate the true Q function by a function whose parameters must be learned. In our case, we employ a network $Q(\boldsymbol{s}, a, \boldsymbol{\omega})$ parametrized by weights $\boldsymbol{\omega}$ to estimate the Q-function $Q(\boldsymbol{s}, a, \boldsymbol{\omega}) \approx Q^*(\boldsymbol{s}, a)$. We minimize the following loss:

$$\mathcal{L}(\boldsymbol{\omega}_i) = \mathbb{E}_{\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{R}_t, \boldsymbol{S}_{t+1}}\Big[(Y_{i-1} - Q(\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{\omega}_i))^2\Big] \tag{4.5}$$

with $Y_{i-1} = R_t + \gamma \max_a (Q(\boldsymbol{S}_{t+1}, a, \boldsymbol{\omega}_{i-1}))$. This may be seen as minimizing the mean squared distance between approximations of the right- and left-hand sides of (4.4). In order to compute (4.5), we sample quadruplets $(\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{R}_t, \boldsymbol{S}_{t+1})$ following the policy implied by Q^{i-1}:

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q(\boldsymbol{s}_t, a, \omega_{i-1}). \tag{4.6}$$

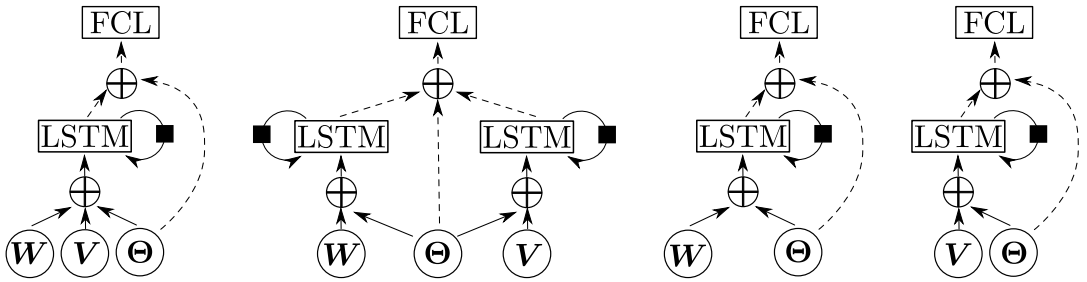
However, instead of sampling only according to (4.6), random actions a_t are taken in ϵ percents of the time steps in order to explore new strategies. This approach is known as epsilon-greedy policy. $\mathcal{L}$ is minimized over $\boldsymbol{\omega}_i$ by stochastic gradient descent. Refer to [81] for more technical details about the training algorithm.

4.3.2 Neural Network Architectures for Q-Learning

The Q-function is modeled by a neural network that takes as input part of the state variable $\boldsymbol{S}_t$, that we define as $\boldsymbol{S}_t^{\Delta t} = \{\boldsymbol{O}_{t-\Delta t} ... \boldsymbol{O}_t\}$. The output is a vector of size $\#\mathcal{A}$ that corresponds to each $Q_\pi(\boldsymbol{s}_t^{\Delta t}, a_t), a_t \in \mathcal{A}$, where $Q_\pi(\boldsymbol{s}_t^{\Delta t}, a_t)$ is built analogously to (4.2). Following [81], the output layer is a fully connected layer (FCL) with linear activations. We propose to use the long short-term memory (LSTM) [49] recurrent neural network to model the Q-function since recurrent neural networks are able to exhibit dynamic behavior for temporal sequences. LSTM are designed such as to prevent the back propagated errors from vanishing or exploding during training. We argue that LSTM is well suited for our task as it is capable of learning temporal dependencies better than other recurrent neural networks or than Markov models. In fact, our model needs to memorize the position

and the motion of the people when the robot turns its head. When a person is not detected anymore, the network should be able to use previous detections back in time in order to predict the direction towards which it should move. Batch normalization is applied to the output of LSTM. The J channels of $\boldsymbol{V}_t$ are flattened before the LSTM layers.

Four different network architectures are described in this section and are evaluated in Section 4.4. In order to evaluate when the two streams of information (audio and video) need to be fused, we propose to compare two architectures: early fusion (*EFNet*) and late fusion (*LFNet*). In early fusion, the audio and visual features are combined into a single representation before modeling time dependencies, e.g. Fig. 4.2(a). Conversely, in late fusion, audio and visual features are modeled separately before fusing them, e.g. Fig. 4.2(b). In order to measure the impact of each modality, we also propose two more network architectures using either audio (*AudNet*) or vision (*VisNet*) information. Fig. 4.2(c) displays *AudNet* and Fig. 4.2(d) displays *VisNet*. Fig. 4.2 employs a compact network representation where time is not explicitly shown, while Fig. 4.3 depicts the unfolded representation of *EFNet* where each node is associated with one particular time instance. Both figures follow the graphical representation used in [42].



(a) *EFNet* **(b)** *LFNet* **(c)** *AudNet* **(d)** *VisNet*

Figure 4.2: Proposed architectures to model the Q-function. Dashed lines indicate connections only used in the last time step. Black squares represent a delay of a single time step. Encircled crosses depict the concatenation of inputs.

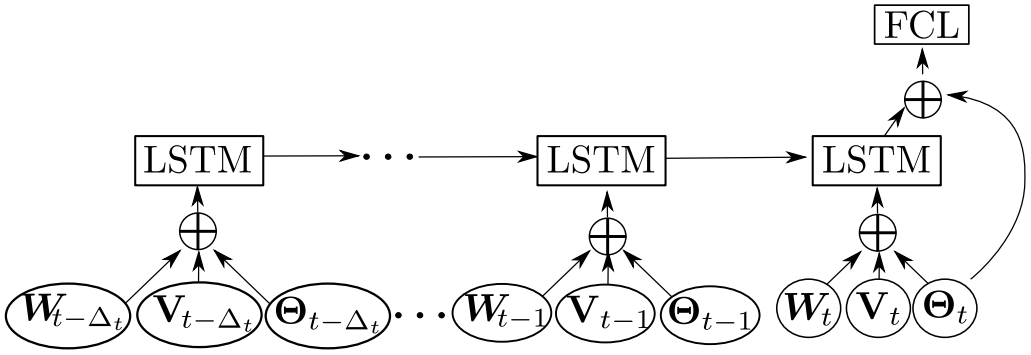


Figure 4.3: Unfolded representation of *EFNet* to better capture the sequential nature of the recurrent model. Encircled crosses depict the concatenation of inputs.

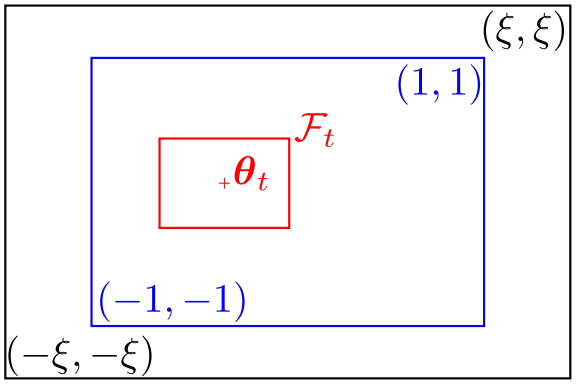


Figure 4.4: Diagram showing all fields used in the proposed simulated environment. The robot's field of view (in red) can move within the reachable field (in blue), whereas the participants can freely move within a larger field (in black).

4.3.3 Pretraining on Simulated Environment

Training from scratch a DQN model can take a long time (in our case ~150000 time steps to converge), and training directly on a robot would not be convenient for two reasons. First, it would entail a long period of training, since each physical action by the robot takes an amount of time that cannot be reduced neither by code optimization nor by increasing our computational capabilities. Second, in the case of HRI, participants would need to move in front of the robot for several hours or days (like in [96]). For these two reasons, we propose to use a transfer learning approach. The Q-function is first learned on a simulated environment, where we simulate people moving and talking, and it is then used to initialize the network employed by the robot. Importantly, the network learned from this simulated environment can be successfully used in the robot without the need of fine-tuning in real data. In this simulated environment, we do not need to generate images and sound signals, but only the observations and rewards the Q-Network receives as input.

We consider that the robot can cover the field $[-1, 1]^2$ by moving its head, but can only visually observe the people within a small rectangular region $\mathcal{F}_t \subset [-1, 1]^2$ centered in position vector $\boldsymbol{\Theta}_t$. The audio observations cover the whole reachable region $[-1, 1]^2$. In each episode, we simulate one or two persons moving with random speeds and accelerations within a field $[-\xi, \xi]^2$ where $\xi > 1$. In other words, people can go to regions that are unreachable for the robot. For each simulated person in the current episode, we consider the position and velocity of their head at time t, $\boldsymbol{h}_t = (u_t^h, v_t^h) \in [-\xi, \xi]^2$ and $\dot{h} = (\dot{u}_t^h, \dot{v}_t^h) \in \mathbb{R}^2$, respectively. At each frame, the person can keep moving, stay without moving, or choose another random direction. The details of the simulated environment generator are given in Algorithm 3. In a real scenario, people can leave the scene so, in order to simulate this phenomenon, we consider two equally probable cases when a person is going out horizontally of the field ($v_t^h \notin [-\xi, \xi]$). In the first case, the person is deleted and instantly recreated on the other side of the field ($v_{t+1}^h = -v_t^h$) keeping the same velocity ($\dot{v}_{t+1}^h = \dot{v}_t^h$). In the second case, the person is going back towards the center ($v_{t+1}^h = v_t^h$ and ($\dot{v}_{t+1}^h = -\dot{v}_t^h$)). A similar approach is used when a person is going out

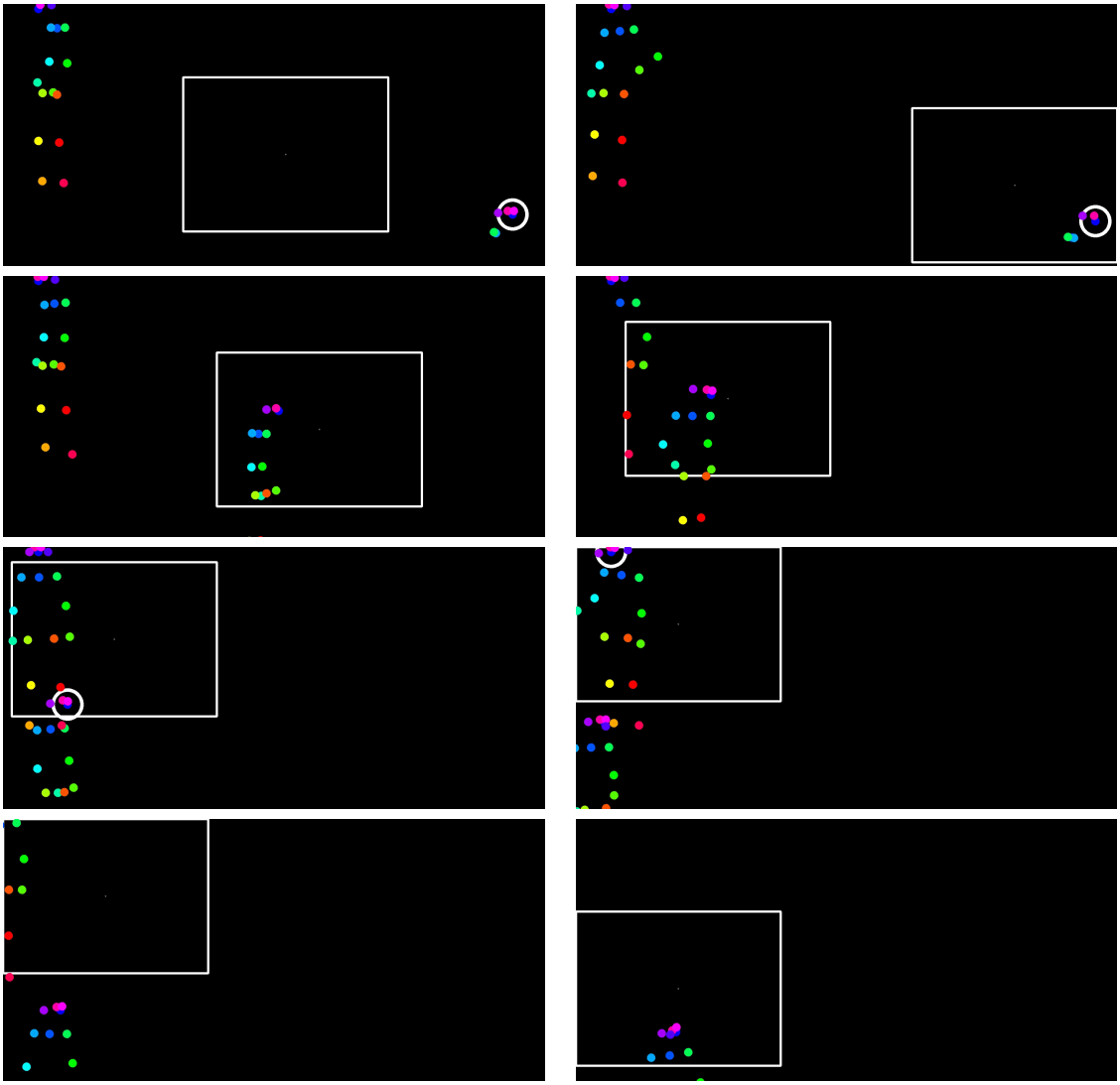


Figure 4.5: Illustrative sequence taken from the simulated environment and employed to pretrain our neural network-based RL approach. The moving square represents the *camera field-of-view* $\mathcal{F}_t$ of the robot. The colored circles represent the joints of a person in the environment. The large white circle represents a person speaking and, therefore, producing speech that can be detected by the speech localization system. Frames are displayed from top to bottom and left to right.

vertically except that we do not create new persons on top of the field because that would imply the unrealistic sudden appearance of new legs within the field. Fig. 4.4 displays a visual representation of the different fields (or areas) defined in our simulated environment, and Fig. 4.5 shows an example of a sequence of frames taken from the simulated environment and used during training.

Moreover, in order to favor tracking abilities, we bias the person motion probabilities such that a person that is faraway from the robot head orientation has a low probability to move, and a person within *camera field-of-view* has a high probability to move. Thus,

when there is nobody in the *camera field-of-view*, the robot cannot simply wait for a person to come in. On the contrary, the robot needs to track the persons that are visible. More precisely, we consider 4 different cases. First, when a person has never been seen by the robot, the person does not move. Second, when a person is in the robot field of view ($\boldsymbol{h}_t \in \mathcal{F}_t$), they move with a probability of 95%. Third, when the person is further than a threshold $\tau \in \mathbb{R}$ from the *camera field-of-view* ($||\boldsymbol{h}_t - \boldsymbol{\Theta}_t||_2 > \tau$), the probability of moving is only 25%. Finally, when the person is not visible but close to the *camera field-of-view* ($||\boldsymbol{h}_t - \boldsymbol{\Theta}_t||_2 < \tau$ and $\boldsymbol{h}_t \notin \mathcal{F}_t$), or when the person is unreachable ($\boldsymbol{h}_t \in [-\xi, \xi] \backslash [-1, 1]$), this probability is 85%. Regarding the simulation of missing detections, we randomly ignore some faces when computing the face features. Concerning the sound modality, we randomly choose between the following cases: 1 person speaking, 2 persons speaking, and nobody speaking. We use a Markov model to enforce continuity in the speaking status of the persons, and we also simulate wrong audio observations.

From, the head position, we need to generate the position of all body joints. To do so, we propose to collect a set $\mathcal{P}$ of poses from an external dataset (the *AVDIAR* dataset [40]). We use a multiple person pose estimator on this dataset and use the detected poses for our simulated environment. This task is not trivial since we need to simulate a realistic and consistent sequence of poses. Applying tracking to the *AVDIAR* videos could provide good pose sequences, but we would suffer from three major drawbacks. First, we would have a tracking error that could affect the quality of the generated sequences. Second, each sequence would have a different and constant size, whereas we would like to simulate sequences without size constraints. Finally, the number of sequences would be relatively limited. In order to tackle these three concerns, we first standardize the output coordinates obtained on *AVDIAR* . Considering the pose p_t^n of the n^{th} person, we sample a subset $\mathcal{P}_t^M \subset \mathcal{P}$ of M poses. Then, we select the closest pose to the current pose: $p_{t+1}^n = \underset{p \in \Pi}{\operatorname{argmin}}\, d(p, p_t^n)$ where

$$d\left(\begin{pmatrix} u_1 \\ v_1 \\ s_1 \end{pmatrix}, \begin{pmatrix} u_2 \\ v_2 \\ s_2 \end{pmatrix}\right) = \frac{1}{\sum_{j=1}^{J} s_1^j s_2^j} \sum_{j=1}^{J} (s_1^j s_2^j) \sqrt{(u_1^j - u_2^j)^2 + (v_1^j - v_2^j)^2} \tag{4.7}$$

This distance is designed to face poses with different number of detected joints. It can be interpreted as an L_2 distance weighted by the number of visible joints in common. The intuition behind this sampling process is that when the size M of $\mathcal{P}_t^M$ increases, the probability of obtaining a pose closer to p_t^n increases. Consequently, the motion variability can be adjusted with the parameter M in order to obtain a natural motion. With this method we can obtain diverse sequences of any size.

4.4 Experiments

4.4.1 Evaluation with a Recorded Dataset

The evaluation of HRI systems is not an easy task. In order to fairly compare different models, we need to train and test the different models on the exact same data. In the

Data: $\mathcal{P}$: a set of poses, δ: time-step
σ: velocity variance, M: pose continuity parameter

Randomly chose N in $[1..3]$.
for $n \in [1..N]$ **do**
  Initialize $(\boldsymbol{h}_0^n, \dot{\boldsymbol{h}}_0^n) \sim \mathcal{U}([-1,1])^2 \times \mathcal{U}([-1,-0.5] \bigcup [0.5,1])^2$.
  Randomly chose p_0^n in $\mathcal{P}$.
end
for $t \in [1..T-1]$ **do**
  for $n \in [1..N]$ **do**
    Randomly chose $motion \in \{Stay, Move\}$
    if $motion = Move$ **then**
      if $\boldsymbol{h}_t^n \notin [-\xi, \xi]^2$ **then**
        The person is leaving the scene.
        See section 4.3.3.
      else
        $\boldsymbol{h}_{t+1}^n \leftarrow \boldsymbol{h}_t^n + \delta(\dot{\boldsymbol{h}}_t^n + \mathcal{N}((0,0), \sigma))$.
        $\dot{\boldsymbol{h}}_{t+1}^n \leftarrow \frac{1}{\delta}(\boldsymbol{h}_{t+1}^n - \boldsymbol{h}_t^n)$
      end
    else
      $\boldsymbol{h}_{t+1}^n \leftarrow \boldsymbol{h}_t^n$
      $\dot{\boldsymbol{h}}_{t+1}^n \sim \mathcal{U}([-1,-0.5] \bigcup [0.5,1])^2$
    end
    Draw $\mathcal{P}_t^M$, a random set of M elements of $\mathcal{P}$
    $p_{t+1}^n \leftarrow \underset{p \in \mathcal{P}_t^M}{\operatorname{argmin}}\, d(p, p_t^n)$
  end
end

Algorithm 3: Generation of simulated moving poses for our simulated environment.

Figure 4.6: Example of a sequence from the *AVDIAR* dataset. The speech direction binary map is superimposed on the image, and the visible landmarks are displayed using a colored skeleton. The camera field of view (in red) is randomly initialized (far left), speech emitted by one of the persons is detected and hence the gaze is controlled (left). The agent is able to get all the persons in the field of view (right), and it gazes at a group of three persons while two other persons move apart (far right).

context of RL and HRI, this is problematic because the data, i.e. what the robot actually sees and hears, depends on the action taken by the robot. Thus, we propose to first evaluate our model with the *AVDIAR* dataset [40]. This dataset was recorded with four

Figure 4.7: Example of a live sequence with two persons. First row shows an overview of the scene, including the participants and the robot. Second row shows the images gathered with the camera mounted onto the robot head. The robot head is first initialized in a position where no face is visible (first column), and the model uses the available landmarks (elbow and wrist) to find the person onto the right (second column). The robot detects the second person by looking around while keeping the first person in its field of view (third column), and gazes the two people walking together (fourth column).

microphones and one high-resolution camera (1920×1080 pixels). These images, due to their wide field of view, are suitable to simulate the motor field of view of the robot. In practical terms, only a small box of the full image simulates the robot's camera field of view. Concerning the observations, we employ visual and audio grids of sizes 7×5 in all our experiments with the *AVDIAR* dataset.

We employ 16 videos for training. The amount of training data is doubled by flipping the video and audio maps. In order to save computation time, the original videos are down-sampled to 1024×640 pixels. The size of the camera field of view where faces can be detected is set to 300×200 pixels using motion steps of 36 pixels each. These dimensions approximately correspond the coverage angle and motion of Nao. At the beginning of each episode, the position of the camera field of view is selected such that it contains no face. We noticed that this initialization procedure favors the exploration abilities of the agent. To avoid a bias due to the initialization procedure, we used the same seed for all our experiments and iterated three times over the 10 test videos (20 when counting the flipped sequences). An action is taken every 5 frames (0.2 seconds).

Fig. 4.6 shows a short sequence of the *AVDIAR* environment, displaying the whole field covered by the *AVDIAR* videos as well as the smaller field of view captured by the robot (the red rectangle in the figure). However, it is important to highlight that transferring the model learned using *AVDIAR* to Nao is problematic and did not work in our preliminary experiments. First, faces are almost always located at the same position (around the image center). Second, all videos are recorded indoors using only two different rooms, and participants are not moving much. Finally, the audio setting is unrealistic for a robotics scenario, e.g. absence of motor noise. Therefore, the main reason for using the *AVDIAR* dataset is to compare our method with other methods in a generic setting.

4.4.2 LIVE EXPERIMENTS WITH NAO

In order to carry out an online evaluation of our method, we performed experiments with a Nao robot. Nao has a 640×480 pixels cameras and four microphones. This robot is particularly well suited for HRI applications because of its design, hardware specifications and affordable cost. Nao's commercially available software can detect people, locate sounds, understand some spoken words, synthesize speech and engage itself in simple and goal-directed dialogs. Our gaze control system is implemented on top of the NAOLab middleware [7] that synchronizes proprioceptive data (motor readings) and sensor information (image sequences and acoustic signals). The reason why we use a middleware is threefold. First, the implementation is platform-independent and, thus, easily portable. Platform-independence is crucial since we employ a transfer learning approach to transfer the model parameters, obtained with the proposed simulated environment, to the Nao software/hardware platform. Second, the use of external computational resources is transparent. This is also a crucial matter in our case, since visual processing is implemented on a GPU which is not available on-board of the robot. Third, the use of middleware makes prototyping much faster. For all these reasons, we employ the remote and modular layer-based middleware architecture named NAOLab. NAOLab consists of four layers: drivers, shared memory, synchronization engine and application programming interface (API). Each layer is divided into three modules devoted to vision, audio and proprioception, respectively. The last layer of NAOLab provides a general programming interface in C++ to handle the sensory data and to manage its actuators. NAOLab provides, at each time step, an image and the direction of the detected sound sources using [68, 69].

We now provide some implementation details specifically related to the Nao implementation. The delay between two successive observations is ~0.3 seconds. The rotating head has a motor field-of-view of 180°. The head motion parameters are chosen such that a single action corresponds to 0.15 radians (~9°) and 0.10 radians (~6°) for horizontal and vertical motions, respectively. Concerning the observations, we employ a visual grid of size 7×5 and an audio grid of size 7×1 in all our experiments with Nao. Indeed, Nao has a planar microphone array and hence sound sources can only be located along the azimuth (horizontal) direction. Therefore the corresponding audio binary map is one-dimensional.

Fig. 4.7 shows an example of a two-person scenario using the *LFNet* architecture. As shown in our recorded experiments [1] , we were able to transfer the exploration and tracking abilities learned using the simulated environment. Our model behaves well independently of the number of participants. The robot is first able to explore the space in order to find people. If only one person is found, the robot follows the person. If the person is static, the robot keeps the previously detected person in the field but keeps exploring the space locally aiming at finding more people. When more people appear, the robot tries to find a position that maximizes the number of people. The main failure cases are related to quick movements of the participants.

[1] A video showing offline and online experiments is available at `https://team.inria.fr/perception/research/deep-rl-for-gaze-control/`

Table 4.1: Comparison of the final reward obtained with different architectures. The best results obtained are displayed in bold.

	AVDIAR				Simulated	
	Face		Speaker		*Face*	*Speaker*
Network	*Training*	*Test*	*Training*	*Test*		
AudNet	1.50 ± 0.03	1.47 ± 0.04	1.92 ± 0.02	1.82 ± 0.03	0.21 ± 0.01	0.33 ± 0.01
VisNet	1.89 ± 0.03	**1.85 ± 0.02**	2.32 ± 0.04	2.23 ± 0.03	0.37 ± 0.04	0.45 ± 0.06
EFNet	1.90 ± 0.03	1.81 ± 0.04	2.40 ± 0.02	2.22 ± 0.03	0.41 ± 0.03	**0.53 ± 0.03**
LFNet	**1.96 ± 0.02**	1.83 ± 0.02	**2.43 ± 0.02**	**2.29 ± 0.02**	**0.42 ± 0.01**	0.52 ± 0.03

4.4.3 Implementation Details

By carefully selecting the resolution used to perform person detection along the method of [19], we were able to obtain visual landmarks in less than 100 ms. Considering that NAOLab gathers images at 10 FPS, this landmark estimator can be considered as fast enough for our purpose. Moreover, [19] follows a bottom-up approach, which allows us to speed-up landmark detection by skipping the costly association step.

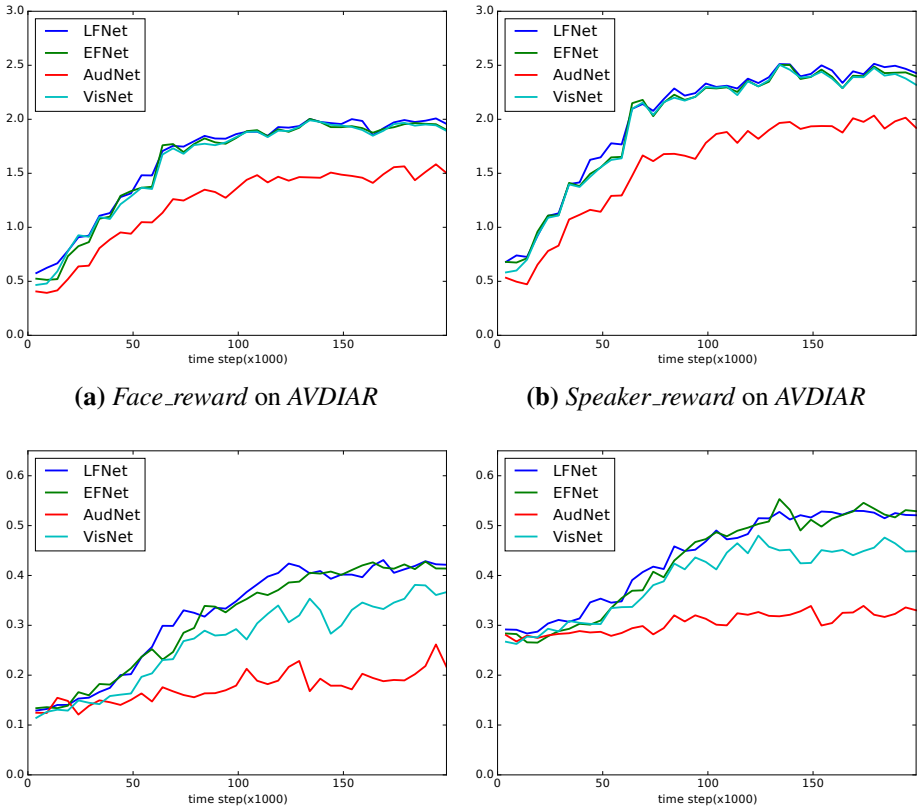
The parameters of our model are based on a preliminary experimentation. We set $\Delta_T = 4$ in all scenarios, such that each decision is based on the last 5 observations. The output size of LSTM is set to 30 (since a larger size does not provide an improvement in performance), and the output size of the FCL is set to 5 (one per action). We use a discount factor (γ) of 0.90. Concerning the training phases, we employed the Adam optimizer [57] and a batch size of 128. In order to help the model to explore the policy space, we use an ϵ-greedy algorithm: while training, a random action is chosen in ϵ% of the cases; we decrease linearly the ϵ value from $\epsilon = 90\%$ to $\epsilon = 10\%$ after 120000 iterations. The models were trained in approximately 45 minutes on both *AVDIAR* and the simulated environment. It is interesting to notice that we obtain this training time without using GPUs. A GPU is only needed for person detection and estimation of visual landmarks (in our case, a Nvidia GTX 1070 GPU).

In the simulated environment, the size of field in which the people can move is set to $\xi = 1.4$. In the case of Nao, the audio observations are provided by the multiple speech-source localization method described in [69].

In all our experiments, we run five times each model and display the mean of five runs to lower the impact of the stochastic training procedure. On *AVDIAR* , the results on both training and test sets are reported in the tables. As described previously, the simulated environment is randomly generated in real time, so there is no need for a separated test set. Consequently, the mean reward over the last 10000 time steps is reported as test score.

4.4.4 Architecture Comparison

In Table 4.1, we compare the final reward obtained while training on the *AVDIAR* dataset and on our simulated environment with the two proposed rewards (*Face_reward* when $\alpha = 0$, and *Speaker_reward* when $\alpha = 1$). Four different networks are tested: *EFNet*,



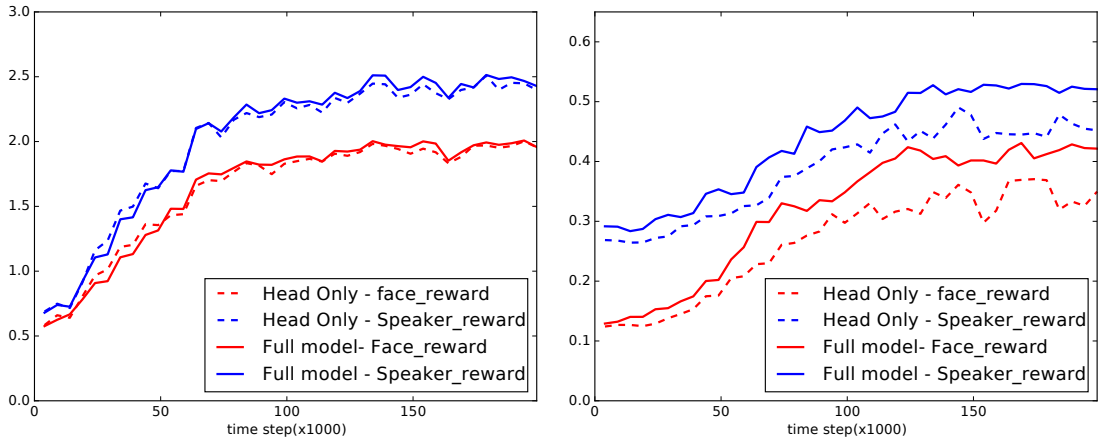
(a) *Face_reward* on *AVDIAR* **(b)** *Speaker_reward* on *AVDIAR*

(c) *Face_reward* on Simulated **(d)** *Speaker_reward* on Simulated

Figure 4.8: Evolution of the reward obtained while training with the two proposed rewards on the *AVDIAR* dataset and on the simulated environment. We average over a 5000 time-step window for a cleaner visualization.

LFNet, *VisNet*, and *AudNet*. The y-axis of Fig. 4.8 shows the average reward per episode, with a clear growing trend as the training time passes (specially in the experiments with the *AVDIAR* dataset), meaning that the agent is learning (improving performance) from experience. On the simulated environment, the best results are indistinctly provided by the late and early fusion strategies (*LFNet* and *EFNet*), showing that our model is able to effectively exploit the complementarity of both modalities. On *AVDIAR*, the late fusion performs slightly better than the early fusion model. Globally, we observe that the rewards we obtain on *AVDIAR* are higher than those obtained on the simulated environment. We suggest two possible reasons. First, the simulated environment has been specifically designed to enforce exploration and tracking abilities. Consequently, it poses a more difficult problem to solve. Second, the number of people in *AVDIAR* is higher (about 4 in average), thus finding a first person to track would be easier. We notice that, on the *AVDIAR* dataset using the *Face_reward*, we obtain a mean reward greater than 1, meaning that, on average, our model can see more than one face per frame. We also observe that *AudNet* is the worst performing approach. However, it performs quite well on *AVDIAR* compared to the simulated environment. This behavior can be explained by the

fact that, on *AVDIAR*, the speech source detector returns a 2D heatmap whereas only the angle is used in the simulated environment. As conclusion, we select *LFNet* to perform experiments on Nao.



(a) *AVDIAR* **(b)** Simulated

Figure 4.9: Evolution of the training reward obtained when using as visual observation the result of either the full-body pose estimation or the face location information.

Fig. 4.9 displays the reward obtained when using only faces as visual observation (dashed lines) in contrast to using the full-body pose estimation (continuous lines). We observe that on the simulated data, the rewards are significantly higher when using the full-body pose estimator. This figure intends to respond empirically to the legitimate question of why a full-body pose estimator is used instead of a simple face detector. From a qualitative point of view, the answer can be found in the type of situations that can solve one and the other. Let's imagine that the robot looks at the legs of a user; in case of using only a face detector, there is no clue that could help the robot to move up its head in order to see a face; however, if a human full-body pose detector is used, the detection of legs implies that there is a torso over them, and a head over the torso. Incidentally, we remark that the difference when using full-body pose and only the head is quite small on the *AVDIAR* dataset. Since this dataset has not been designed to be challenging for this task, an explanation lies in the fact that the tilt angle required to see the faces is almost always the same. Then, the robot moves its head to until it reaches the best tilt angle, whether it has seen legs or not.

4.4.5 Parameter Study

In this section, we describe the experiments devoted to evaluate the impact of some of the principal parameters involved. More precisely, the impact of three parameters is analyzed. First, we compare different values for the discount factor γ that defines the importance of short-term rewards as opposed to long-term ones (see Section 4.3). Second, we compare different window sizes. It corresponds to the number of past observations that are

Table 4.2: Comparison of the final reward obtained using different discounted factors (γ). The mean and standard deviation over 5 runs are reported. The best average results obtained are displayed in bold.

	AVDIAR		*Simulated*
γ	Training	Test	
25	**1.96 ± 0.02**	1.85 ± 0.02	0.33 ± 0.09
50	**1.96 ± 0.02**	**1.86 ± 0.03**	0.35 ± 0.08
75	**1.96 ± 0.02**	1.85 ± 0.02	**0.43 ± 0.11**
90	1.94 ± 0.02	1.83 ± 0.02	0.42 ± 0.12
99	1.95 ± 0.01	1.84 ± 0.02	0.42 ± 0.12

used to make a decision (see Section 4.3.2). Finally, we compare different sizes for the LSTM network that is employed in all our proposed architectures (see Section 4.3.2). It corresponds to the dimension of the cell state and hidden state that are propagate by the LSTM.

In Table 4.2, different discount factors are compared. With *AVDIAR*, high discount factors are prone to overfit as the difference in performance between training and test is large. With the simulated environment, low discount values perform worse because the agent needs to perform several actions to detect a face, as the environment is rather complex. Consequently, a model that is able to take into account future benefits of each action performs better. Finally, in Table 4.3, we compare different LSTM sizes. We observe that increasing the size does not lead to significantly better results, which is an interesting outcome since, from a practical point of view, smaller LSTMs faster the training.

Different window sizes are compared in Table 4.4. We can conclude that the worst results are obtained when only the current observation is used (window size of 1). We also observe that, on *AVDIAR* , the model performs well even with short window lengths (2 and 3). In turn, with a more complex environment, as the proposed simulated environment, a longer window length tends to perform better. We interpret that using a larger window size helps the network to ignore the noisy observations and to remember the position of people that left the field of view. We report the training time for each window length. We observe that, using a smaller time window speeds up training since it avoids back-propagating the gradient deeply in the LSTM network.

Table 4.3: Comparison of the final reward obtained using different LSTM sizes. The mean and standard deviation over 5 runs are reported. The best average results obtained are displayed in bold.

LSTM size	*AVDIAR*		*Simulated*
	Training	Test	
30	**1.96 ± 0.01**	1.85 ± 0.03	0.42 ± 0.11
60	1.95 ± 0.02	1.86 ± 0.02	**0.43 ± 0.12**
120	1.92 ± 0.04	**1.87 ± 0.02**	0.41 ± 0.10

Table 4.4: Comparison of the final reward obtained using different window lengths (Δ_T). The mean and standard deviation over 5 runs are reported. The best average results obtained are displayed in bold. The training time is reported for each configuration.

	AVDIAR			*Simulated*	
$\Delta_T + 1$	Training	Test	Time(s$\times 10^3$)	Test	Time(s$\times 10^3$)
1	1.92 ± 0.03	1.82 ± 0.03	3.05 ± 0.22	0.26 ± 0.04	3.07 ± 0.15
2	1.94 ± 0.02	$\mathbf{1.85 \pm 0.02}$	2.25 ± 0.99	0.36 ± 0.04	3.09 ± 0.17
3	1.93 ± 0.01	1.84 ± 0.01	2.95 ± 0.38	0.42 ± 0.02	2.98 ± 0.27
5	1.94 ± 0.02	1.84 ± 0.02	3.30 ± 0.46	$\mathbf{0.43 \pm 0.01}$	3.40 ± 0.14
10	1.94 ± 0.02	1.84 ± 0.02	2.05 ± 0.22	0.40 ± 0.02	3.85 ± 0.36
20	$\mathbf{1.96 \pm 0.01}$	1.82 ± 0.02	3.00 ± 0.00	0.42 ± 0.02	5.35 ± 0.36
128	1.94 ± 0.02	1.82 ± 0.03	18.90 ± 0.77	0.41 ± 0.03	52.98 ± 5.23

4.4.6 Comparison with the State of the Art

We perform a comparative evaluation with the state of the art. To the best of our knowledge, no existing work addresses the problem of finding an optimal gaze policy in the HRI context. In [13] a heuristic that uses an audio-visual input to detect, track and involve multiple interacting persons is proposed. Hence we compare our learned policy with their algorithm. On the simulated environment, as the speech source is only localized in the azimuthal plane (see section 4.4.2), we randomly gaze along the vertical axis in order to detect faces. In [12] two strategies are proposed to evaluate visually controlled head movements. A first strategy consists of following a person and rotating the robot head in order to align the person's face with the image center. A second strategy consists in randomly jumping every 3 seconds between persons. Obviously, the second strategy was designed as a toy experiment and does not correspond to a natural behavior. Therefore, we compare our RL approach with their first strategy. Unfortunately, the case where nobody is in the field of view is not considered in [12]. To be able to compare their method in the more general scenario addressed here, we propose the following handcrafted policy in the case no face is detected in the visual field of view: (i) *Rand*: A random action is chosen; (ii) *Center*: Go towards the center of the *acoustic field-of-view*; (iii) *Body*: If a limb is detected, the action $\uparrow$ is chosen in order to find the corresponding head, otherwise, *Rand* is followed, and (iv) *Audio*: Go towards the position of the last detected speaker.

Importantly, in our model the motor speed is limited, since the robot can only select unitary actions. When implementing other methods, one could argue that this speed limitation is inherent to our approach and that other methods may not suffer from it. However, it is not realistic to consider that the head can move between two opposite locations of the auditory field in two consecutive frames with an infinite speed. Therefore, we report two scores, first using the same speed value than the one used in our model (referred to as *equal*), and second by making the unrealistic assumption that the motor speed is infinite (referred to as *infinite*). This second evaluation protocol is therefore biased towards handcrafted methods. The results are reported in Table 4.5.

First, we notice that none of the handcrafted methods can compete with ours when

Table 4.5: Comparison of the final rewards obtained with different handcrafted policies. The performances of competitor methods are reported considering the two speed assumptions (*equal*/*infinite*) described in the text.

	AVDIAR		Simulated	
	Face_reward	*Speaker_reward*	*Face_reward*	*Speaker_reward*
Ban et al.[12]+*Rand*	1.19/1.21	1.45/1.59	0.25/0.26	0.40/0.37
Ban et al.[12]+*Center*	1.62/1.68	1.95/2.01	0.14/0.11	0.28/0.29
Ban et al.[12]+*Body*	1.23/1.20	1.40/1.52	0.27/0.26	0.39/0.37
Ban et al.[12]+*Audio*	1.54/1.63	1.84/2.06	0.32/0.39	0.43/0.48
Bennewitz et al.[13]	1.56/1.55	2.07/2.05	0.30/**0.42**	0.35/0.50
LFNet	$\mathbf{1.83 \pm 0.02}$	$\mathbf{2.29 \pm 0.02}$	$\mathbf{0.42 \pm 0.01}$	$\mathbf{0.52 \pm 0.03}$

considering the same motor speed. On both environments, *LFNet* largely outperforms all handcrafted models. This clearly justifies policy learning and the use of RL for gaze control. Concerning [12], *Center* obtains the best result among the [12]'s variances on *AVDIAR* and the worst on Simulated according to the *Face_reward* metric. This can be explained by the fact that, as mentioned in Section 4.4.1, most persons are located around the image center and, therefore, this dummy strategy works better than more sophisticated ones. A similar behavior can be observed with the *Speaker_reward* metric. We observe that in both environments using audio information, when no face is detected, improves the performance with respect to *Rand*. The second best performance on *AVDIAR* is obtained by [13] with *Speaker_reward*. On the simulated environment, [13] equals the score obtained by our proposal when making the unrealistic assumption of infinite motor speed. In that case, [13] is marginally inferior to our proposal according to the *Speaker_reward*. When considering equal speed limit, our RL approach significantly outperforms the handcrafted policy of [13] (26% and 48% higher according to *Face_reward* and *Speaker_reward*, respectively). All these results highlight the crucial importance of audio-visual fusion in the framework of RL and in the context of gaze control.

4.5 CONCLUSIONS

In this chapter, we presented a neural network-based reinforcement learning approach to solve the gaze robot control problem. In particular, our agent is able to autonomously learn how to find people in the environment by maximizing the number of people present in its field of view while favoring people that speak. A simulated environment is used for pre-training prior to transfer learning to a real environment. Neither external sensors nor human intervention are necessary to compute the reward. Several architectures and rewards are compared on three different environments: two offline (real and simulated datasets) and real experiments using a robot. Our results suggest that combining audio and visual information leads to the best performance, as well as that pre-training on simulated data can even make unnecessary to train on real data. By thoroughly experimenting on a publicly available dataset and with a robot, we provide empirical evidence that our RL approach outperforms handcrafted strategies.

This approach shows an original adaptive solution for a complex problem, yet its ap-

plication in actual scenarios may still be limited. The action space is too restrictive for a real usage with simultaneously low speed and unnatural robot head movement. Future work may need to introduce head rotation velocity, and/or a continuous action space, and may require changing the reinforcement learning paradigm used. On another note, the robot objective, as defined by the reward, is to find groups of people. This behavior may lead to maximizes the social information (still debatable, for instance in the case of a presenter speaking to an audience), but can appear unnatural, *e.g.* when the robot head looks like it points in between two persons. Even though this framework allows to pre-train the behavior of a robot that may then continue to adapt in situ, many details must be carefully designed to fit the desired scenario. Instead, future work could be centered on actual applications, and may combine this approach with inverse reinforcement learning [4].

CHAPTER 5

CONCLUSION

5.1 SUMMARY

There are countless processes involved in a social interaction. Some cues are emitted and perceived, consciously or not, and it helps communicate more efficiently. A social humanoid robot must be able to detect and analyze cues that are valuable for understanding what is currently happening. In parallel, the robot can reproduce human-like cues to optimize the communication process. In this thesis, we studied how a robot that participates in a social interaction can infer to whom or what each person is looking at, *i.e.* their Visual Focus of Attention (VFOA). We encountered three sub-problems that we addressed using data-driven approaches, either employing existing datasets or simulated data.

First, we addressed frame-by-frame VFOA inference during a social interaction, in a favorable setting, *i.e.* knowing the location of objects of interest and being able to evaluate people head poses. Our formulation uses a switching linear dynamical system to characterize the dependency between VFOA, gaze direction and head pose. It leads to an online inference and a training algorithm. Results are competitive with the state of the art, and the method still works when directly transferred to a new dataset.

Second, we address the more realistic setting in which the locations of objects of interest are not known a priori. By using a top-view representation to model the region of interaction, we propose to estimate the location of objects of interest from a sequence of head poses. The mapping is done by training a convolutional encoder/decoder on simulated data. Tests on both simulated data and a real dataset show the interest of learning-based approaches. In a wider sense, we demonstrated that the detection of out-of-view objects from gaze following is a challenging problem.

Third, we focused on controlling the robot gaze direction to have people in the robot's field of view. Indeed, previous tasks require to see people's head within the robot camera image. We achieve an efficient gaze control strategy using reinforcement learning. The robot can autonomously evaluate its reward from audio and visual observations and does not need human supervision. The audio-visual observations are associated to a head

movement action using an LSTM Recurrent neural network. A synthetic environment is used to speed up training, and the weights of the network are then transferred to a real Nao robot that is able to successfully find and follow people. Extensive experiments on synthetic data and on a recorded dataset show the advantage of our learning-based approach comparatively to handcrafted strategies.

5.2 FUTURE RESEARCH DIRECTIONS

In this section, we present several promising research directions for improving each task separately. Later, we suggest some perspectives for combining them or solving related problems using the presented methodologies.

In chapter 2 and 3, gaze direction is not directly used but replaced with head orientation. The reason for this is that eye-gaze estimators tend to be unreliable with non-frontal faces. In particular, all publicly available eye-gaze datasets are composed of near-frontal faces and/or are only labeled with on-screen gaze following. The design and recording of an unconstrained gaze direction dataset would provide a better basis for gaze-based scene understanding methods. However, it is difficult to create a dataset including 3D gaze direction labels. A small dataset could be recorded by filming people looking at objects whose location is known a priori. This dataset could potentially be augmented with synthetically generated images, *e.g.* using [51, 131]. The methods that rely on head pose instead of gaze direction can be improved by means of gaze direction with some minor adaptations.

In chapter 3, several simplifying hypotheses have been adopted to release the constraints commonly employed on gaze following problems. First, the elevation coordinate from 3D location has been discarded. However, there is no conceptual reason why a similar formulation replacing 2D heat-maps with 3D ones, and adding one dimension to the convolutions, would not achieve similar results (at the cost of significantly increasing computing time). Moreover, the grid discretization has advantages, *e.g.* invariance to the number of people and objects, but also drawbacks, *e.g.* it is not applicable to unbounded environments. Finally, a benchmark of other spatial models and their constraints for this problem would be valuable.

In chapter 4, the robotic agent learns to move its head so that the camera is directed towards the maximal number of people, favoring people who speak. When the field of view is too narrow to see everyone at once, isolated persons tend to be ignored. Looking alternatively all people to know everybody's location with high confidence is a different problem that requires a specific modeling. In particular, an audio-visual identity model should be introduced and remain consistent over time (as in [12]). The hurdle comes from introducing a person-wise representation into a method that is independent from the number of people. Preliminary experiments have been done as reported in appendix B. Independently, results from group psychology (*e.g.* F-formations [27, 109]) could be introduced as prior information to improve the model capabilities over a pure learning-based approach.

It is important to note that the three aforementioned contributions employ different frameworks that can hardly be combined together. It is possible to develop a software that uses them in a sequential pipeline. The robot controls its gaze to find people; people in the field of view are used to estimate locations of objects of interest; when objects' locations are known, the frame-wise VFOA can be computed. However, this pipeline completely ignores the complementarity of the three tasks, since jointly solving the tasks can benefit all of them. For instance, people often look at each other, so an object of interest out of the field of view may actually be another person that the robot should look at. On the other hand, we can compute in parallel the estimated locations of objects of interest and VFOAs given those estimates. This would check that the proposed locations are meaningful. Combining the three tasks in a single framework would require rethinking their formulation, but could provide far better results than optimizing each one separately.

The detection of out-of-view objects in chapter 3 cannot use vision analysis since, by definition, objects are *outside* the image. However, robotic gaze control, as in chapter 4, can be used to turn the camera towards a person's VFOA. The gaze direction from the initial image, before turning the camera, could be combined with a saliency algorithm on the final image (as in [100]) to estimate the location of the object of interest, and possibly guess what it is. This could be referred as gaze-based active saliency.

Finally, an important milestone for the work presented in this thesis (and its potential extensions) would be the design of a scenario in which the robot has to autonomously interact with unguided people in unrehearsed social situations. This would validate the robot skills for each task, and improve fault tolerance in the pipeline. An example of such a scenario has been designed in [115], in which they focus on slightly different problems. For our tasks, a possible scenario could be based on the *Vernissage* setup [54], adapted to take place in a real museum with actual visitors.

5.3 REFLECTIONS

During a PhD thesis in computer science, there is little or no incentive to think about societal and moral applications of the research. I suspect this is not different for other fields such as physics or chemistry. In this section, I want to detail some commonly expressed issues and open some prospects. Please note that this is not sociological research, but rather a short discussion over some interesting questions related to the thesis.

The study of non-verbal cues, like gaze direction as reported in this manuscript, has applications in human-robot interaction. Nonetheless, since a lot of non-verbal cues are unconsciously emitted or perceived, it is possible to use them for manipulative purposes. For instance, the message in an advertisement can be received in a completely different manner depending on the visual display. Of course, companies and political movements are aware of this in their communication strategy. An interactive advertisement could analyze the person currently looking at it, and formulate the message to maximize the person's receptiveness. In this context, the gaze trajectory can help to figure out the

importance hierarchy of objects in the image for the observer, and model his/her state of mind.

More widely, there are many concerns on Artificial Intelligence (AI) since its widespread usage from digital companies raises questions. The most debated topic concerns privacy. Indeed, machine learning, the sub-field of AI responsible for most recent breakthrough, requires training data. In many fields, data are sensitive yet necessary to progress. For instance, some medical conditions, political opinions, or criminal records are associated with social stigmas and should not be made publicly available. Additionally, data anonymization is often not reliable enough [86]. Moreover, databases must be secured to avoid leaks. Since security measures often come at the expense of usability, there may be huge security breaches in many computer systems. The high number of security flaws discovered in connected devices is a recent example [134]. Also, even when there are legitimate reasons to collect some data, it is questionable to use them for another task later, without asking again for the permission of the concerned people. Legislation has adapted to protect individual privacy, but may only be selectively enforced. In particular, security and privacy sometimes directly contradict each other, and authorities may be inclined to favor security, with the risk of power abuse *e.g.* against political opponents.

Besides the data-related concerns, it should be noted that AI is a very efficient and versatile technology and some applications are debatable. First, AI-controlled weapons (sometimes called intelligent weapons) blur the moral responsibility associated to killing, and are feared by public opinions, as it could be turned against the populations. Another application, for which the issues are a bit sneakier, is the use of AI to assist human decisions. Indeed, the use of AI to help taking good decision based on data may reproduce human bias and discrimination, without hindsight. For example, in the US, a judiciary decision assisting program tend to overestimate recidivism of black people [22], strengthening their social determinism. Finally, society recently realized that social networks – using AI-based algorithms – can be used to manipulate the public opinion. More widely, covertly manipulating the population by capitalizing on the widespread use of data-based algorithms in our ultra-connected world may be the fastest path towards a totalitarian system (similar to the ones described in *A brave new world* by Aldous Huxley, *Nineteen Eighty-Four* by Georges Orwell, or recently *La zone du dehors* by Alain Damasio). More generally, AI is a powerful technology, and similarly to other technologies, *e.g.* nuclear fission, it is certainly not intrinsically dangerous. Problems come when it is used by careless, malicious and/or insatiable people.

A common fear in the general public is the concept of a sentient machine that chooses to harm humans or to take control of the society. In the same category, the technological singularity [106] is the hypothetical advent of a machine that surpasses human intelligence. It is thus able to create an even more powerful machine, recursively leading to an "intelligence explosion". I believe these scenarios still belong to the realm of science-fiction. Indeed, as far as I know, there is no recent peer-reviewed scientific paper claiming to know how or when a sentient machine could be achieved. This should not prevent us from thinking about what would happen if the time comes, and a lot of fiction stories propose a reflection on these topics (Remarkable examples include *Frankenstein or the*

modern Prometheus by Mary Shelley, the *Robot* series by Isaac Asimov, *2001, A Space Odyssey* by Stanley Kubrick and Arthur C. Clarke or also *Blade Runner* by Ridley Scott from an original novel by Philip K. Dick). However, as explained above, non-sentient machines already provide concerning issues.

On a different note, an underestimated matter of AI is its energy consumption. Indeed, deep learning requires massive computation on energy demanding graphic cards, and their use is getting more and more widespread. In a climate changing world, it should be reminded that electricity is still massively generated using fossil fuels. Obviously, other activities like crypto-currency mining are extremely energy expensive, but it should not exempt AI users and creators from taking into account the energy consumption factor.

As a conclusion, there are many activities that benefit and will benefit from Artificial Intelligence in the upcoming years [112]. Yet there are downsides, and researchers are too seldom prompted to question moral and ethical aspects, or to stand up against objectionable uses of their scientific findings.

CHAPTER A

ADDITIONAL MATERIAL CHAPTER 2

A.1 VFOA TRANSITION PROBABILITIES

Using the notations introduced in Section 2.3.3, let i, $1 \leq i \leq N$, be an active target. In Section 2.3.3 we showed that in practice the entries of the probability transition matrix can have up to 15 different expressions. For completeness, these expressions are listed below.

- The VFOA of i at $t-1$ is neither an active nor a passive target ($k = 0$):

$$p_1 = P(\mathrm{V}_t^i = 0 | \mathrm{V}_{t-1}^i = 0)$$
$$p_2 = P(\mathrm{V}_t^i = j | \mathrm{V}_{t-1}^i = 0)$$

- The VFOA of i at $t-1$ is a passive target ($N < k \leq N + M$):

$$p_3 = P(\mathrm{V}_t^i = 0 | \mathrm{V}_{t-1}^i = k)$$
$$p_4 = P(\mathrm{V}_t^i = k | \mathrm{V}_{t-1}^i = k)$$
$$p_5 = P(\mathrm{V}_t^i = j | \mathrm{V}_{t-1}^i = k)$$

- The VFOA of i at $t-1$ is an active target ($1 \leq k \leq N, k \neq i$):

$$
\begin{aligned}
p_6 &= P(\mathrm{V}_t^i = 0 | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = 0) \\
p_7 &= P(\mathrm{V}_t^i = k | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = 0) \\
p_8 &= P(\mathrm{V}_t^i = j | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = 0) \\
p_9 &= P(\mathrm{V}_t^i = 0 | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = i) \\
p_{10} &= P(\mathrm{V}_t^i = k | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = i) \\
p_{11} &= P(\mathrm{V}_t^i = j | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = i) \\
p_{12} &= P(\mathrm{V}_t^i = 0 | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = l) \\
p_{13} &= P(\mathrm{V}_t^i = k | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = l) \\
p_{14} &= P(\mathrm{V}_t^i = l | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = l) \\
p_{15} &= P(\mathrm{V}_t^i = j | \mathrm{V}_{t-1}^i = k, \mathrm{V}_{t-1}^k = l)
\end{aligned}
$$

A.2 VFOA Learning

This appendix provides the formulae allowing to estimate the 15 transitions probabilities as explained in Section 2.5.1.

$$
\hat{p}_1 = \frac{\sum_{q=1}^{Q} \sum_{i=2}^{N_q} \sum_{t=2}^{T_q} \delta_0(\mathrm{V}_t^{q,i}) \delta_0(\mathrm{V}_{t-1}^{q,i})}{\sum_{q=1}^{Q} \sum_{i=2}^{N_q} \sum_{t=2}^{T_q} \delta_0(\mathrm{V}_{t-1}^{q,i})}
$$

$$
\hat{p}_2 = \frac{\sum_{q=1}^{Q} \sum_{i=2}^{N_q} \sum_{t=2}^{T_q} \sum_{j \neq i} \delta_j(\mathrm{V}_t^{q,i}) \delta_0(\mathrm{V}_{t-1}^{q,i})}{\sum_{q=1}^{Q} \sum_{i=2}^{N_q} \sum_{t=2}^{T_q} \delta_0(\mathrm{V}_{t-1}^{q,i})}
$$

$$
\hat{p}_3 = \frac{\sum_{q=1}^{Q} \sum_{i=2}^{N_q} \sum_{t=2}^{T_q} \sum_{k=N_q+1}^{N_q+M_q} \delta_0(\mathrm{V}_t^{q,i}) \delta_k(\mathrm{V}_{t-1}^{q,i})}{\sum_{q=1}^{Q} \sum_{i=2}^{N_q} \sum_{t=2}^{T_q} \sum_{k=N_q+1}^{N_q+M_q} \delta_k(\mathrm{V}_{t-1}^{q,i})}
$$

$$\hat{p}_4 = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{k=N_q+1}^{N_q+M_q}\delta_k(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{k=N_q+1}^{N_q+M_q}\delta_k(\mathrm{V}_{t-1}^{q,i})}$$

$$\hat{p}_5 = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{k=N_q+1}^{N_q+M_q}\sum_{j\neq i,k}\delta_j(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{k=N_q+1}^{N_q+M_q}\delta_k(\mathrm{V}_{t-1}^{q,i})}$$

$$\hat{p}_6 = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_0(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_0(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_0(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_7 = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_0(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_0(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_8 = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{j\neq i,k}\delta_j(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_0(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_0(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_9 = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_0(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_i(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_i(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_{10} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_i(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_i(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_{11} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{j\neq i,k}\delta_j(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_i(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_i(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_{12} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_0(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_{13} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_k(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_{14} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_l(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}$$

$$\hat{p}_{15} = \frac{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\sum_{j\neq i,k,l}\delta_j(\mathrm{V}_t^{q,i})\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}{\sum_{q=1}^{Q}\sum_{i=2}^{N_q}\sum_{t=2}^{T_q}\sum_{\substack{k=1\\k\neq i}}^{N_q}\sum_{l\neq i,k}\delta_k(\mathrm{V}_{t-1}^{q,i})\delta_l(\mathrm{V}_{t-1}^{q,k})}$$

CHAPTER B

TEACHING, INTERNSHIP

During the second and third years of the PhD, I taught computer science and mathematics at Grenoble IUT2 [1], under the supervision of Francis Brunet-Manquat. Grenoble IUT2 is an institute that delivers job-centered diplomas. Most students are registered to the two-years training, corresponding to the first two years of the license/bachelor (L1 and L2). Additionally, students can candidate to a third year (L3) specialization that leads to a license degree. In parallel, a special one-year training course called "Année spéciale" (AS) is available to non-computer science students that want to switch their field of study. I taught the following courses over two years to students from various levels:

- Advanced database design (24h, L2)
- Android programming (24h, L2 + 24h, AS)
- Distributed systems (16h, L3)
- Discrete mathematics (32h, L1)

After the three year funding ended, I decided to apply to a one-year teaching assistant position called ATER (Attaché Temporaire d'Enseignement et de Recherche) at Grenoble INP Ensimag [2]. ATER positions are especially designed for last year PhD students. Ensimag is a french "École d'ingénieur" in mathematics and computer science. It proposes three-years training courses; the corresponding academic years are the end of license (L3), and master (M1 and M2). I taught the following courses in one year:

- Introduction to programming (22h, L3)
- Algorithmic and data structures (33h, L3)
- Object-oriented programming (36h, M1)

[1] https://iut2.univ-grenoble-alpes.fr/en/
[2] http://www.grenoble-inp.fr/welcome/

- Statistics (19h L3)
- Formal languages (14h, L3)
- Algorithmic and programming, Refresher course (32h, M1)
- Software engineering project in C (28h, L3)

Each year, approximately 100 students register to first year at IUT2, and 250 at Ensimag. Since it is obviously too large for good tutoring, I taught to one or two groups of 25-30 students per subject. Different teachers taught to the other groups. To ensure the homogeneity among courses, most of the teaching materials had been made by each course main supervisor. Yet in all courses, besides teaching, I had to grade my students. This either consisted in marking exams, evaluating student projects, or sometimes both. Additionally, I participated in the making of the exams, and in some courses, I updated or created some teaching materials. In total, during the PhD, I gave around 300 hours of lectures.

Finally, I co-advised with Dr. Radu Horaud the internship of Victor Bros during summer 2018. Victor, an Ensimag student, worked on the extension of the autonomous gaze control from chapter 4. He adapted the method to work on another robot, and started some experiments to give the robot a better representation of people position based on audio-visual tracking [12]. Results are promising for the future but are not mature enough to be integrated into this manuscript.

REFERENCES

[1] Henny Admoni and Brian Scassellati. Social eye gaze in human-robot interaction: a review. *Journal of Human-Robot Interaction*, 6(1):25–63, 2017.

[2] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1988.

[3] Michael J. Arcaro, Peter F. Schade, Justin L. Vincent, Carlos R. Ponce, and Margaret S. Livingstone. Seeing faces is necessary for face-domain formation. *Nature Neuroscience*, 20(10):1404–1412, 2017.

[4] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *arXiv preprint arXiv:1806.06877*, 2018.

[5] Stylianos Asteriadis, Kostas Karpouzis, and Stefanos Kollias. Visual focus of attention in non-calibrated environments using gaze estimation. *International Journal of Computer Vision*, 107, 2014.

[6] Sileye O Ba and Jean-Marc Odobez. Recognizing visual focus of attention from head pose in natural meetings. *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, 2009.

[7] Fabien Badeig, Quentin Pelorson, Soraya Arias, Vincent Drouard, Israel Gebru, Xiaofei Li, Georgios Evangelidis, and Radu Horaud. A distributed architecture for interacting with nao. In *ACM ICMI*, pages 385–386, 2015.

[8] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI*, 2017.

[9] Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *IEEE CVPR*, 2014.

[10] Ruzena Bajcsy. Active perception. Technical report, University of Pennsylvania, 1988.

[11] Dare A Baldwin. Understanding the link between joint attention and language. *Joint attention: Its origins and role in development*, pages 131–158, 1995.

[12] Yutong Ban, Xavier Alameda-Pineda, Fabien Badeig, Sileye Ba, and Radu Horaud. Tracking a varying number of people with a visually-controlled robotic head. In *IEEE/RSJ IROS*, 2017.

[13] M. Bennewitz, F. Faber, D. Joho, M. Schreiber, and S. Behnke. Towards a humanoid museum guide robot that interacts with multiple persons. In *IEEE-RAS*, pages 418–423, 2005.

[14] C M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[15] Ali Borji, Daniel Parks, and Laurent Itti. Complementary effects of gaze direction and early saliency in guiding fixations during free viewing. *Journal of vision*, 2014.

[16] Ernesto Brau, Jinyan Guan, Tanya Jeffries, and Kobus Barnard. Multiple-gaze geometry: Inferring novel 3d locations from gazes observed in monocular video. In *ECCV*, 2018.

[17] Rechele Brooks and Andrew N Meltzoff. The development of gaze following and its relation to language. *Developmental science*, 8(6):535–543, 2005.

[18] Judee K Burgoon, Deborah A Coker, and Ray A Coker. Communicative effects of gaze behavior: A test of two contrasting explanations. *Human Communication Research*, 12(4):495–524, 1986.

[19] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *IEEE CVPR*, 2017.

[20] Isarun Chamveha, Yusuke Sugano, Daisuke Sugimura, Teera Siriteerakul, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Head direction estimation from low resolution images with scene adaptation. *Computer Vision and Image Understanding*, 117, 2013.

[21] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *International Journal of Robotics Research*, 30(11):1343–1377, 2011.

[22] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 2017.

[23] Joon Son Chung, Andrew W Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. In *CVPR*, pages 3444–3453, 2017.

[24] M. Cohen, I. Shimshoni, E. Rivlin, and A. Adam. Detecting Mutual Awareness Events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[25] Alexis Conneau, Holger Schwenk, Loıc Barrault, and Yann Lecun. Very deep convolutional networks for natural language processing. *arXiv preprint*, 2016.

[26] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[27] Marco Cristani, Loris Bazzani, Giulia Paggetti, Andrea Fossati, Diego Tosato, Alessio Del Bue, Gloria Menegaz, and Vittorio Murino. Social interaction discovery by statistical analysis of f-formations. In *BMVC*, volume 2, page 4, 2011.

[28] Francisco Cruz, German I Parisi, Johannes Twiefel, and Stefan Wermter. Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario. In *IEEE/RSJ IROS*, pages 759–766, 2016.

[29] Joris Domhof, Aswin Chandarr, Maja Rudinac, and Pieter Jonker. Multimodal joint visual attention model for natural human-robot interaction in domestic environments. In *IROS*, 2015.

[30] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE CVPR*, 2015.

[31] Vincent Drouard, Radu Horaud, Antoine Deleforge, Silèye Ba, and Georgios Evangelidis. Robust head-pose estimation based on partially-latent mixture of linear regressions. *IEEE Transactions on Image Processing*, 26, January 2017.

[32] S. Duffner and C. Garcia. Visual focus of attention estimation with unsupervised incremental learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2015.

[33] K Duncan and S Sarkar. Saliency in images and video: a brief survey. *IET Computer Vision*, 6(6):514–523, 2012.

[34] Nathan J Emery. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience & Biobehavioral Reviews*, 2000.

[35] Lifeng Fan, Yixin Chen, Ping Wei, Wenguan Wang, and Song-Chun Zhu. Inferring shared attention in social scene videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6460–6468, 2018.

[36] John M Findlay and Iain D Gilchrist. *Active vision: The psychology of looking and seeing*. Number 37 in Oxford Psychology. Oxford University Press, 2003.

[37] E. G. Freedman and D. L. Sparks. Eye-head coordination during head-unrestrained gaze shifts in rhesus monkeys. *Journal of Neurophysiology*, 1997.

[38] Edward G Freedman. Coordination of the eyes and head during visual orienting. *Experimental Brain Research*, 190, 2008.

[39] Alexandra Frischen, Andrew P Bayliss, and Steven P Tipper. Gaze cueing of attention: visual attention, social cognition, and individual differences. *Psychological bulletin*, 133(4):694, 2007.

[40] I. Gebru, S. Ba, X. Li, and R. Horaud. Audio-visual speaker diarization based on spatiotemporal bayesian fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[41] Ali Ghadirzadeh, Judith Bütepage, Atsuto Maki, Danica Kragic, and Mårten Björkman. A sensorimotor reinforcement learning framework for physical Human-Robot Interaction. In *IEEE/RSJ IROS*, pages 2682–2688, 2016.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[43] Michael A. Goodrich and Alan C. Schultz. Human-robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[44] Hieronymus HLM Goossens and AJ Van Opstal. Human eye-head coordination in two dimensions under different sensorimotor conditions. *Experimental Brain Research*, 114, 1997.

[45] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jerome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. The nao humanoid: a combination of performance and affordability. *CoRR abs/0807.3223*, 2008.

[46] Daniel Guitton and Michel Volle. Gaze control in humans: eye-head coordination during orienting movements to targets within and beyond the oculomotor range. *Journal of neurophysiology*, 58(3):427–459, 1987.

[47] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, March 2010.

[48] Hongsheng He, Shuzhi Sam Ge, and Zhengchen Zhang. Visual attention prediction using saliency determination of scene understanding for social robots. *International Journal of Social Robotics*, 2011.

[49] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[50] A. K. A. Hong, J. Pelz, and J. Cockburn. Lightweight, low-cost, side-mounted mobile eye tracking system. In *IEEE WNYIPW*, 2012.

[51] Yibo Hu, Xiang Wu, Bing Yu, Ran He, and Zhenan Sun. Pose-guided photorealistic face rotation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[52] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[53] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2001.

[54] Dinesh Babu Jayagopi, Samira Sheiki, David Klotz, Johannes Wienke, Jean-Marc Odobez, Sebastien Wrede, Vasil Khalidov, Laurent Nyugen, Britta Wrede, and Daniel Gatica-Perez. The vernissage corpus: A conversational human-robot-interaction dataset. In *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, pages 149–150. IEEE, 2013.

[55] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE TPAMI*, 2013.

[56] Tatsuya Kawahara, Takuma Iwatate, and Katsuya Takanashi. Prediction of turn-taking by combining prosodic and eye-gaze information in poster conversations. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[57] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[58] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *IJRR*, 2013.

[59] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *IEEE CVPR*, June 2016.

[60] Krzysztof Krejtz, Cezary Biele, Dominik Chrzastowski, Agata Kopacz, Anna Niedzielska, Piotr Toczyski, and Andrew Duchowski. Gaze-controlled gaming: immersive and difficult but not cognitively overloading. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1123–1129. ACM, 2014.

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[62] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf. Visual analytics for mobile eye tracking. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):301–310, Jan 2017.

[63] Pablo Lanillos, João Filipe Ferreira, and Jorge Dias. A bayesian hierarchy for robust gaze estimation in human-robot interaction. *International Journal of Approximate Reasoning*, 87, 05 2017.

[64] Stéphane Lathuilière, Benoit Massé, Pablo Mesejo, and Radu Horaud. Deep reinforcement learning for audio-visual gaze control. In *IROS*, 2018.

[65] Stéphane Lathuilière, Benoit Massé, Pablo Mesejo, and Radu Horaud. Neural network based reinforcement learning for audio-visual gaze control in human-robot interaction. *Pattern Recognition Letters*, 2018.

[66] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5325–5334, 2015.

[67] X. Li, L. Girin, R. Horaud, and S. Gannot. Multiple-speaker localization based on direct-path features and likelihood maximization with spatial sparsity regularization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1997–2012, Oct 2017.

[68] Xiaofei Li, Laurent Girin, Fabien Badeig, and Radu Horaud. Reverberant sound localization with a robot head based on direct-path relative transfer function. In *IEEE/RSJ IROS*, 2016.

[69] Xiaofei Li, Laurent Girin, Radu Horaud, and Sharon Gannot. Multiple-speaker localization based on direct-path features and likelihood maximization with spatial sparsity regularization. *IEEE/ACM TASLP*, 2017.

[70] Huiying Liu, Min Xu, Jinqiao Wang, Tianrong Rao, and Ian Burnett. Improving visual saliency computing with emotion intensity. *IEEE IEEE Trans. Neural Netw. Learn. Syst.*, 2016.

[71] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.

[72] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[73] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Adaptive linear regression for appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, Oct 2014.

[74] Feng Lu, Takahiro Okabe, Yusuke Sugano, and Yoichi Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 32, 2014.

[75] Päivi Majaranta and Andreas Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*, pages 39–65. Springer, 2014.

[76] M. J. Marin-Jimenez, A. Zisserman, M. Eichner, and V. Ferrari. Detecting people looking at each other in videos. *IJCV*, 2014.

[77] Benoit Massé, Silèye Ba, and Radu Horaud. Simultaneous estimation of gaze direction and visual focus of attention for multi-person-to-robot interaction. In *IEEE ICME*, Seattle, WA, July 2016.

[78] Benoit Massé, Silèye Ba, and Radu Horaud. Tracking gaze and visual focus of attention of people involved in social interaction. *IEEE TPAMI*, 2017.

[79] Y. Matsumoto, T. Ogasawara, and A. Zelinsky. Behavior recognition based on head pose and gaze direction measurement. In *IEEE IROS*, volume 3, 2000.

[80] Noriaki Mitsunaga, Christian Smith, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. *JRSJ*, 2006.

[81] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[82] Hans Moravec. *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.

[83] Sankha S Mukherjee and Neil Martin Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, 17(11):2094–2107, 2015.

[84] K. P. Murphy. Switching Kalman filters. Technical report, UC Berkeley, 1998.

[85] E. Murphy-Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 2009.

[86] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[87] David G Novick, Brian Hansen, and Karen Ward. Coordinating turn-taking with gaze. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 3, pages 1888–1891. IEEE, 1996.

[88] T. Ohno and N. Mukawa. A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In *Proceedings of the ETRA Symposium*. ACM, 2004.

[89] K. Otsuka, J. Yamato, and Y. Takemae. Conversation scene analysis with dynamic bayesian network based on visual head tracking. In *IEEE ICME*, 2006.

[90] Daniel Parks, Ali Borji, and Laurent Itti. Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision research*, 2015.

[91] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

[92] A. Patron-Perez, M. Marszałek, A. Zisserman, and I. D. Reid. High five: Recognising human interactions in TV shows. In *British Machine Vision Conference*, 2010.

[93] Roberto Pinillos, Samuel Marcos, Raul Feliz, Eduardo Zalama, and Jaime Gómez-García-Bermejo. Long-term assessment of a service robot in a hotel environment. *Robotics and Autonomous Systems*, 79:40–57, 2016.

[94] S. Pourmehr, J. Thomas, J. Bruce, J. Wawerla, and R. Vaughan. Robust sensor fusion for finding HRI partners in a crowd. In *IEEE ICRA*, pages 3272–3278, 2017.

[95] Zhen Qin and Christian R Shelton. Social grouping for multi-target tracking and head pose estimation in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 2016.

[96] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro. Robot gains social intelligence through multimodal deep reinforcement learning. In *IEEE Humanoids*, pages 745–751, 2016.

[97] Ahmed Hussain Qureshi, Yutaka Nakamura, Yuichiro Yoshikawa, and Hiroshi Ishiguro. Show, attend and interact: Perceivable human-robot social interaction through neural attention Q-network. In *IEEE ICRA*, pages 1639–1645, 2017.

[98] Anoop Kolar Rajagopal, Ramanathan Subramanian, Elisa Ricci, Radu L Vieriu, Oswald Lanz, and Nicu Sebe. Exploring transfer learning approaches for head pose classification from multi-view surveillance images. *International Journal of Computer Vision*, 109, 2014.

[99] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? In *NIPS*, 2015.

[100] Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. Following gaze in video. In *CVPR*, 2017.

[101] Ben Robins, Kerstin Dautenhahn, R Te Boekhorst, and Aude Billard. Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills? *Universal Access in the Information Society*, 4(2):105–120, 2005.

[102] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 2015.

[103] Edward Rosbergen, Rik Pieters, and Michel Wedel. Visual attention to advertising: A segment-level analysis. *Journal of consumer research*, 24(3):305–314, 1997.

[104] M. Rothbucher, C. Denk, and K. Diepold. Robotic gaze control using reinforcement learning. In *IEEE HAVE*, 2012.

[105] Dmitry Rudoy, Dan B Goldman, Eli Shechtman, and Lihi Zelnik-Manor. Learning video saliency from human gaze using candidate selection. In *IEEE CVPR*, 2013.

[106] Anders Sandberg. An overview of models of technological singularity. In *Roadmaps to AGI and the Future of AGI Workshop, Lugano, Switzerland, March*, volume 8. Citeseer, 2010.

[107] Boris Schauerte and Rainer Stiefelhagen. "Look at this!" Learning to guide visual saliency in human-robot interaction. In *IROS*, 2014.

[108] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[109] Francesco Setti, Chris Russell, Chiara Bassetti, and Marco Cristani. F-formation detection: Individuating free-standing conversational groups in images. *PloS one*, 10(5):e0123783, 2015.

[110] S. Sheikhi and J-M. Odobez. Recognizing the visual focus of attention for human robot interaction. In *Human Behavior Understanding Workshop*, 2012.

[111] Samira Sheikhi and Jean-Marc Odobez. Combining dynamic head pose–gaze mapping with the robot conversational state for attention recognition in human–robot interactions. *Pattern Recognition Letters*, 66, 2015.

[112] Thomas B Sheridan. Human–robot interaction: status and challenges. *Human factors*, 58(4):525–532, 2016.

[113] D. Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *Control Theory Applications, IET*, 2010.

[114] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[115] Gabriel Skantze, Anna Hjalmarsson, and Catharine Oertel. Turn-taking, feedback and joint attention in situated human-robot interaction. *Speech Communication*, 65:50–66, 2014.

[116] P. Smith, M. Shah, and N. Da Vitoria Lobo. Determining driver visual attention with one camera. *IEEE Transactions on Intelligent Transportation Systems*, 4, 2003.

[117] Hyun Soo Park and Jianbo Shi. Social saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4777–4785, 2015.

[118] J. S. Stahl. Amplitude of human head movements associated with horizontal saccades. *Experimental Brain Research*, 126, 1999.

[119] R. Stiefelhagen and J. Zhu. Head orientation and gaze direction in meetings. In *Human Factors in Computing Systems*, 2002.

[120] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 1st edition, 1998.

[121] Ryu Takeda and Kazunori Komatani. Sound source localization based on deep neural networks with directional activate function exploiting phase information. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 405–409. IEEE, 2016.

[122] Andrea L Thomaz, Guy Hoffman, and Cynthia Breazeal. Reinforcement learning with human teachers: Understanding how people want to teach robots. In *IEEE RO-MAN*, pages 352–357, 2006.

[123] Takumi Toyama, Thomas Kieninger, Faisal Shafait, and Andreas Dengel. Gaze guided object recognition using a head-mounted eye tracker. In *Proceedings of the ETRA Symposium*, 2012.

[124] Stefan Treue. Visual attention: the where, what, how and why of saliency. *Current opinion in neurobiology*, 13(4):428–432, 2003.

[125] M. Vázquez, A. Steinfeld, and S. E. Hudson. Maintaining awareness of the focus of attention of a conversation: A robot-centric reinforcement learning approach. In *IEEE RO-MAN*, 2016.

[126] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, volume 1, 2001.

[127] Wenguan Wang and Jianbing Shen. Deep visual attention prediction. *IEEE TIP*, 2018.

[128] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Mach. Learn.*, 1992.

[129] Ping Wei, Yang Liu, Tianmin Shu, Nanning Zheng, and Song-Chun Zhu. Where and why are they looking? jointly inferring human attention and intentions in complex tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[130] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn*, 1992.

[131] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Gazedirector: Fully articulated eye gaze redirection in video. In *Computer Graphics Forum*, 2018.

[132] Yan Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, Oswald Lanz, and Nicu Sebe. A multi-task learning framework for head pose estimation under target motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 2016.

[133] L. H. Yu and M. Eizenman. A new methodology for determining point-of-gaze in head-mounted eye tracking systems. *IEEE Transactions on Biomedical Engineering*, 51, Oct 2004.

[134] Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal, and Chenren Xu. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, HotNets-XIV, New York, NY, USA, 2015. ACM.

[135] Z. Yucel, A. A. Salah, C. Mericli, T. Mericli, R. Valenti, and T. Gevers. Joint attention by gaze interpolation and saliency. *IEEE Transactions on System Men and Cybernetics. Part B.*, 2013.

[136] Sang-Seok Yun. A gaze control of socially interactive robots in multiple-person interaction. *Robotica*, 35(11):2122–2138, 2017.

[137] Xenophon Zabulis, Thomas Sarmis, and Antonis A Argyros. 3D head pose estimation from multiple distant views. In *BMVC*, 2009.

[138] Hao Zhang, Alexander C Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2126–2136. IEEE, 2006.

[139] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2015.

[140] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2299–2308. IEEE, 2017.