EPFL

Thèse n° 7284

Accurate Nod and 3D Gaze Estimation for Social Interaction Analysis

Présentée le 25 mars 2020

à la Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Yu YU

Acceptée sur proposition du jury

Prof. J.-Ph. Thiran, président du jury
Dr. J.-M. Odobez, directeur de thèse
Prof. M. Valstar, rapporteur
Prof. Y. Sugano, rapporteur
Prof. O. Hilliges, rapporteur

École polytechnique fédérale de Lausanne

2020

To my family and friends...

Acknowledgements

The four years' PhD life is one of my most cherished memories. It is not so easy to describe the past four years I spent. It can be happy and fulfilled. It can also be stressful and frustrated sometimes. However, I am so lucky to study in such a quiet and beautiful place like Martigny, and get surrounded by so many nice and helpful colleagues. Here, I would like to thank them all for their help and support.

First of all, I would like to thank my supervisor, Jean-Marc. Your expertise, experience and patient guidance is the key to the success of my thesis. I would always remember the time we spent together to discuss the idea, do the experiment and finish the paper. I believe your inspiration and instruction would have a long lasting impact on my life. I also want to thank Prof. Jean-Philippe Thiran, Prof. Otmar Hilliges, Prof. Yusuke Sugano and Prof. Michel Valstar for kindly taking part of my thesis jury.

I am grateful to the Swiss National Science Foundation (SNSF) for its financial support to my PhD study through UBIMPRESSED project of its Sinergia interdisciplinary program. I also thank the financial support from the MuMMER project of the European Unions Horizon 2020 research and innovation programme.

I also want to express my gratitude to the colleagues I worked closely with: Kenneth, Gang Liu, Remy, Yiqiang, Skanda, Laurent and Di Wu. It is my honour to work with you and thanks for all the experience, ideas and inspirations you gave me to finish my PhD. I am also happy to be a member of Perception and Activity Understanding Group. The group meeting, group lunch and group activity are always full of fun. Thank to all the past and current members: Michael, Weipeng, Angel, Remy, Gulcan, Yiqiang, Kenneth, Alex, Rui Hu, Di Wu, Gang Liu, Yuanzhouhan, Nam.

Idiap is nice place to study and conduct research. I thank all the Idiap employees for their efforts. Special thanks to Nadine and Sylvie, they are always patient to provide kind help in lots of stuffs.

Besides, I also appreciate the 4 months I work as a research intern at SenseTime where Rui Zhao, Dapeng Chen, Suichan Li and Bo Zhao taught me lots of knowledge in other research areas. Thanks for the idea exchanging and work collaboration.

Lastly, I would like to thank my parents and my wife for your accompany, encouragement and

love.

Martigny, January 12, 2020 Yu Yu

Abstract

Non-verbal behaviours play an important role in human communication since it can indicate human attention, serve as communication cue in interactions, or even reveal higher level personal constructs. For instance, head nod, a common non-verbal behaviour, can express the agreement or emphasis when people are listening or speaking. Besides, gaze, another non-verbal behaviour, conveys the human attention and can even provide access to thought processes. With the development of Internet and multimedia, large amount of vision data including videos and images becomes accessible and there are more and more requests on video analysis of human behaviour. Therefore, it is meaningful and important to develop vision based methods to extract non-verbal behaviours automatically.

In this thesis, we attempt to address the recognition of two subtle while important non-verbal behaviours, head nod and gaze. The task of head nod detection is to identify a head movement where the head is rotating up and down along the sagittal plane one or several times while the task of gaze estimation is to infer the 3D Line of Sight with respect to a World Coordinate System. Both tasks have already found applications in areas like Psychology and Sociology (social analysis by head nod detection, mental health care by analyzing gaze), Human Computer and Human Robot Interaction (behaviour recognition or integration to enable smooth interaction), Virtual Reality (rendering improvement accounting for the user's gaze directions).

To address these two problems, we first investigated the task of head pose estimation which is a fundamental task for both head nod detection and gaze estimation. We proposed HeadFusion, an approach for 360° robust head pose tracking. Basically, this is a model based method which relies on depth information. It mainly addresses the weakness of 3D morphable model (3DMM) based methods which usually require frontal or mid-profile poses since the 3DMM model only cover the face region. Our approach, however, achieves a complete head representation by combining the strengths of a 3DMM model fitted online with a prior-free reconstruction of a 3D full head model providing support for pose estimation from any viewpoint. In addition, we also proposes a symmetry regularizer for accurate 3DMM fitting under partial observations, and exploit visual tracking to address natural head dynamics with fast accelerations. Extensive experiments show that our method achieves accurate and robust head pose tracking in difficult scenarios.

Based on the estimated head pose, we designed a head nod detection approach. Compared to

previous approaches, two contributions are made: i) the head rotation dynamic is computed within the head coordinate instead of the camera coordinate, leading to pose invariant gesture dynamics; ii) besides the rotation parameters, a feature related to the head rotation axis is proposed so that nod-like false positives due to body movements could be eliminated. The experiments demonstrate the robustness of our approach.

We then change our research focus to gaze estimation. To achieve robust remote gaze sensing, we first explore the application of multitask learning on gaze estimation. Concretely, we introduce a Constrained Landmark-Gaze Model (CLGM) modelling the joint variation of eye landmark locations (including the iris center) and gaze directions. By relating explicitly visual information (landmarks) to the more abstract gaze values, we demonstrate that the estimator is more accurate and easier to learn. Our framework also decouples gaze estimation from irrelevant geometric variations in the eye image (scale, translation) thanks to a CLGM based decoder.

Lastly, we address the problem of person-specific gaze model adaptation from only a few reference training samples. The main and novel idea is to improve gaze adaptation by generating additional training samples through gaze redirection. Our contributions here are threefold: (i) the proposed gaze redirection framework is based on synthetic data which provides aligned training pairs to predict accurate inverse mapping fields; (ii) a self-supervised approach is designed for domain adaptation; (iii) we exploit the gaze redirection samples to improve the person-specific gaze estimation. Extensive experiments show the validity of our gaze retargeting on person-specific gaze estimation.

Contents

1 Introduction

Non-verbal behaviours play an important role in people's daily communication, since they can convey meaning, thought or personality information which may not be expressed by verbal means. For instance, head nod, a very common non-verbal behaviour, is frequently performed by people when expressing agreement or emphasizing something. Due to this importance and the prevalence of multimedia technology and large datasets, there is a growing interest for the video analysis of human behaviour. In this new context, we believe the extraction and analysis of non-verbal human behaviour is significant and can be applied into many contexts. In social interaction scenarios, detection of non-verbal behaviour can be used to analyze whether a person is actively involved in an activity or whether the person is having a pleasant interaction, and these information can be provided back to help improve one's social skills. In Human-Robot-Interaction (HRI) setting, on one hand, non-verbal behaviours such as hand gestures can be performed by human to give orders to robot. On the other hand, non-verbal behaviours can be integrated into robot's action so that the HRI can be made more natural and friendly. For example, Andrist et al. [2014] attempts to adapt human gaze aversions to a conversational robot to make the conversation between human and robot more natural. Therefore, my research goal is to visually recognize and extract non-verbal behaviours of human and apply them to social interaction analysis.

Among the various non-verbal behaviours, head nod and eye gaze are two important behaviours. Both of them are subtle signals but can reflect the attention, activeness or higher level personal constructs. Due to the rich information they can convey, in recent years there has been a growing interest on head nod detection and gaze estimation. In particular, with the development of deep learning and the availability of large datasets, more and more accurate and robust approaches are reported. However, despite the significant advances achieved, the current vision based head nod detection and gaze estimation methods are still facing many challenges in real life application.

This thesis aims to develop accurate head nod and 3D gaze estimation (inferring the 3D Line of Sight in a World Coordinate System) methods for social interaction analysis. We expect our approach to be applied in non-constrained real life situations by addressing key challenges

like extreme head pose, fast head motion, low resolution imaging and diversity of factors related to the variability in human appearance. In the rest of this introductory chapter we will present relevant applications to further motivate the need for head nod and gaze estimation. Then, the problem of automatic head nod and gaze estimation will be formally defined along with their challenges. We will then state the main goals of this thesis and list its contributions. Finally, an overview of the thesis will be provided.

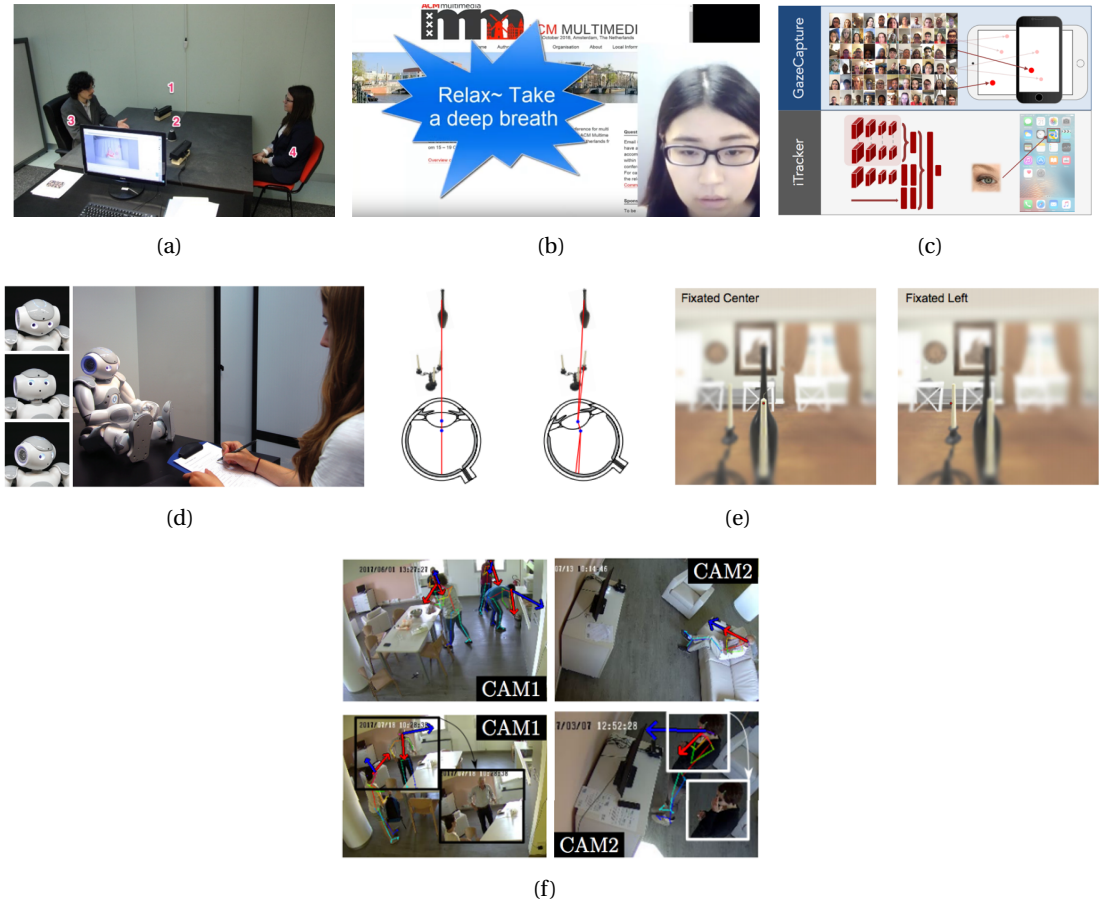
Please note that besides head nod, other head gestures such as head shake can also express important and explicit social signals. Furthermore, we have already collected a database (Q & A scenario) and implemented a detector for both head shake and head nod (not so accurate as a nod detector alone though). However, compared with other head gestures, we believe the head nod is a more frequently performed gesture and can convey more diverse information besides communication signals, e.g. the activeness of person, which meets the requirement of UBImpressed Project for first impression study. Therefore, this thesis mainly explore the problem of head nod detection for head gesture part and we leave the multi-head-gesture detection as a future work.

1.1 Motivation

There are many applications and fields of study which can benefit from automatic nonverbal behaviour extraction, and in particular from the estimation of nods and gaze. Below we list some of them.

Psychology and Sociology. Nonverbal behaviours like head motion and gaze play active roles in human communication. For instance, head nod is often used in face-to-face conversation and has semantic functions. For listeners, they mainly nod to signal yes to a question, or show their interest, agreement and approval to the information they receive. Other functions may include enhancing communicative attention, or anticipating an attempt to capture the floor by occurring in synchrony with the others speech as conversational feedback [Hadar et al., 1983; J.Allwood and L.Cerrato., 2003], along with other cues like gaze [Ba and Odobez, 2008]. For speakers, they usually perform nods to emphasize their speech and in general convey the feeling of conviction or excitement. Therefore, researchers in the field of sociology study have been using these non-verbal cues to analyze human's perceived social variables and predict first impressions [Muralidhar et al., 2016; Finnerty et al., 2016; Muralidhar et al., 2017] (Fig. 1.1a). Besides, as pointed out in [Wetherby et al., 2014], young autistic children tend to have different gaze patterns than normal children. Therefore, non-verbal behaviours like gaze behaviour is also exploited in psychology areas like mental health care [Vidal et al., 2012] and stress analysis [Huang et al., 2016] (Fig. 1.1b).

Human Computer and Human Robot Interaction. In Human Computer or Robot Interaction settings, on one hand it is important for a machine to understand human's non-verbal behaviours so that proper actions can be made as responses [Shi et al., 2019; Rudenko et al.,



(a) (b) (c)

(d) (e)

(f)

Figure 1.1: Applications of non-verbal behaviours. (a) First impression prediction based on RGBD sensors [Muralidhar et al., 2016]. (b) Stress analysis [Huang et al., 2016]. (c) Gaze interaction on mobile devices [Krafka et al., 2016]. (d) Non-verbal behaviour integration in human robot interaction [Andrist et al., 2014]. (e) Visual rendering improvement using gaze [Konrad et al., 2019]. (f) Gaze interaction in assisted living environments [Dias et al., 2019].

2019]. On the other hand, the non-verbal behaviours can be integrated into the behaviour of a robot [Andrist et al., 2014] (Fig. 1.1d) or a virtual agent [Oertel et al., 2016] to enable a smooth and pleasant interaction. With the development of mobile technology in recent years, some works nowadays also used human gaze to improve the interaction between human and mobile phones [Krafka et al., 2016; Müller et al., 2019] (Fig. 1.1c).

Virtual Reality. With the development of Virtual Reality (VR) industry in recent years, there is a growing interest on accurate gaze estimation. By inferring the human attention from gaze, not only the quality of visual rendering can be improved [Konrad et al., 2019; Chen et al., 2019]

(Fig. 1.1e), but also the computational cost of graphic hardware would be reduced.

Assisted Systems. There are several ways in which non-verbal behaviour recognition can be applied in Assisted Systems. One typical example is assisted driving. For instance, by monitoring the gaze, the inattentive behaviours of drivers can be detected to improve the driving safety [Naqvi et al., 2018]. Besides, non-verbal behaviour recognition like gaze is also used in assisted living environments to infer how a person interacts with the environment [Dias et al., 2019] (Fig. 1.1f).

Interface Design. By perceiving the human attention when they look at an interface (e.g. the page of web or software), it is possible to evaluate the property and significance of the displayed visual elements [Itoh et al., 2019] and further guide the design or rearrangement of these elements.

1.2 Problem Definition and Challenges

The primary goal of our thesis is to detect head nod and gaze. However, to achieve that, a first step is to detect and track people's face or head with high accuracy and robustness, which is not trivial. Therefore in the following, we introduce three main problems we have addressed in our work.

1.2.1 Head Pose Estimation

Shown in Fig. 1.2, the head pose can be described by three rotation angles of orthogonal directions, pitch, yaw and roll, which correspond to the rotation around the x, y and z-axis respectively. The task of head pose estimation is to estimate the three angles from an observed face or head.

However, visually estimating head poses accurately and robustly is difficult due to problems such as human shape and appearance variability, extreme head poses, facial expressions, the non-rigid nature of the face, and illumination variations. The development of consumer 3D RGB-D sensors offers an alternative solution. Instead of only providing 2D observations in which fundamental information is lost after projection, the 3D sensor measures the depth information that is inherently required for 3D head pose estimation. Although better performance on head pose estimation has been reported with the help of depth sensors [Fanelli et al., 2011; Papazov et al., 2015], most works only consider applications where the subject's face is nearly frontal. Being able to track the head uninterruptedly in non-constrained natural scenarios where unexpected cases such as fast motions, occlusions, and more profile or adverse poses are presented, remains an open problem.

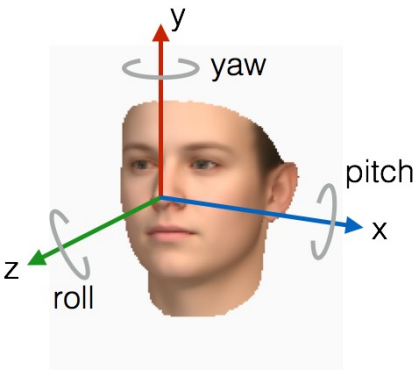


Figure 1.2: Rotation angles along the axes of a head coordinate system. Head nods mainly correspond to the up and down rotation around the x-axis.

1.2.2 Head Nod Detection

Head gestures are a series of head rotations performed around the neck. Among them, a head nod is the movement where the head is mainly rotating up and down along the pitch direction one or several times. Therefore, the task of head nod detection is to identify whether a perceived person is performing such head movements at a specific moment.

To detect head nod, many existing literatures rely on a head tracker to estimate the head pose first and then extract motion dynamics from the head poses [Nakamura et al., 2013; A.Kapoor and R.Picard, 2001]. Besides the challenge of head pose estimation mentioned above, it is also difficult to extract precise motion dynamics from head poses. On one hand, in real and non-constrained scenario, the head nods can be very subtle and not explicit. In such situation, the extracted dynamics would be noisy. On the other hand, it is often assumed that interlocutors have a similar head pose (usually frontal head pose) in training and testing data. Therefore, achieving camera pose invariant dynamics or features is an unexplored challenge.

1.2.3 Gaze Estimation

The problem of remote gaze estimation can be categorized into 3 classes (shown in Fig. 1.3):

- **Gaze fixation point estimation** is to estimate the 2D fixation point of human gaze [Krafka et al., 2016] on a flat surface (screen, tablet). It is usually applied in mobile devices;
- **Gaze object estimation** targets at retrieving the object (or position in an image) people are looking at [Recasens et al., 2015]. This task is usually termed as **gaze following**;
- **3D gaze direction estimation** is to infer the 3D Line of Sight (LoS, the ray pointing out from the fovea and passing through the corneal nodal point, usually expressed by a pitch angle and a yaw angle) in a World Coordinate System (WCS) [Funes-Mora and Odobez, 2016]. We denote this task as **3D gaze estimation** in the rest of this thesis.

In this thesis, we mainly focus on the problem of 3D gaze estimation. In addition, the higher

Figure 1.3: Gaze estimation problems. (a) 3D gaze estimation (gaze direction estimation). [Funes-Mora and Odobez, 2016, 2012]. (b) Gaze fixation point estimation. [Krafka et al., 2016]. (c) Gaze following (gaze object estimation). [Recasens et al., 2015]

level topic of gaze behaviour recognition [Siegfried et al., 2019] is not discussed in this thesis.

Traditional gaze estimation methods include model-based geometrical methods and appearance based methods. The former are more accurate, but the techniques used so far to extract eye landmarks (eye corners, iris) require high resolution images (limiting the freedom of motion) and relatively open eyes since the gaze is estimated from sparse features which are most often detected in a separate task. Appearance based methods directly infer the gaze vector from eye images. They have been shown to be more robust to eye resolution or gazing direction (*e.g.* looking down with eyelid occlusion) variabilities. Thus, this is not surprising that recent works have explored this inference problem via deep regression [Zhang et al., 2016, 2017; Krafka et al., 2016; Fischer et al., 2018; Liu et al., 2018; Park et al., 2018].

Nevertheless, although progress has been reported, direct regression of gaze still suffers from several limitations:

- **Lack of data.** The sizes of benchmark gaze datasets [Funes Mora et al., 2014; Zhang et al., 2016; Smith et al., 2013] are relatively small compared to other vision tasks like image classification, since accurate gaze annotation is complex and expensive. To address the lack of data, domain adaptation methods [Shrivastava et al., 2017] have proposed to use synthetic images for training, but completely eliminating the domain discrepancies between real and synthetic eye images is hard.

- **Systematic bias.** Existing gaze datasets usually use different gaze coordinate systems and data pre-processing methods, in particular for geometric normalization (rectification) relying on different head pose estimators. This introduces a between-dataset systematic bias regarding the gaze ground truth [Yu et al., 2018a].
- **Eye cropping.** An accurate and unified eye cropping is difficult to achieve in real application. This means the size and location of the eye regions may significantly vary in the cropped eye images, due to bad eye/landmark localization, or when changing datasets. Since gaze estimation is very sensitive to the subtle relative positions and shapes of eye landmarks, such variations can significantly alter the gaze estimation outcomes. Though data augmentation can partially handle this problem, an explicit model of this step may improve the generalization ability to new datasets, imperfect cropping, or new eyes.
- **Person-specific bias.** Liu et al. [2018] legitimately argue that gaze can not be fully estimated from the visual appearance since the alignment difference between the optical axis (the line connecting the eyeball center and the pupil center) and the visual axis (the line connecting the fovea and the nodal point [Funes Mora and Odobez, 2014]) is person specific, and vary within -2 to 2 degrees across the population. Therefore, it is not optimal to train a single generic model for accurate cross-person gaze estimation.

1.3 Objective and Thesis Contributions

The main objective of this thesis is to achieve accurate and robust extraction of two non-verbal behaviours, head nod and gaze, under non-constrained free setting and minimal user interventions. To fulfil this goal, we made the following contributions.

360° **Robust Head Pose Tracking.** As mentioned above, head pose estimation is a fundamental task for both head nod detection and gaze estimation. Although 3D morphable model (3DMM) based methods relying on depth information usually achieve accurate results, they usually require frontal or mid-profile poses which precludes a large set of applications where such conditions can not be guaranteed, like monitoring natural interactions from fixed sensors placed in the environment. A major reason is that 3DMM models usually only cover the face region. In this thesis, we presented a framework named HeadFusion which combines the strengths of a 3DMM model fitted online with a prior-free reconstruction of a 3D full head model providing support for pose estimation from any viewpoint. In addition, we also proposes a symmetry regularizer for accurate 3DMM fitting under partial observations, and exploit visual tracking to address natural head dynamics with fast accelerations. Extensive experiments show that our method achieves state-of-the-art performance on the public BIWI dataset, as well as accurate and robust results on UbiPose, an annotated dataset of natural interactions that we make public and where adverse poses, occlusions or fast motions regularly occur.

This work has been published in [Yu et al., 2017, 2018b].

Head Pose and Body Motion Invariant Nod Detector. We proposed a novel nod detection approach based on a full 3D face centered rotation model. Compared to previous approaches, we make two contributions. Firstly, the head rotation dynamic is computed within the head coordinate instead of the camera coordinate, leading to pose invariant gesture dynamics. Secondly, besides the rotation parameters, a feature related to the head rotation axis is proposed so that nod-like false positives due to body movements could be eliminated. The experiments on two-party and four-party conversations demonstrate the validity of the approach.

This work has been published in [Chen et al., 2015]. Based on the internship work of Yiqiang Chen, I conducted further experiments on the proposed approach and made thorough comparison with different settings. In addition, I also implemented an online system for head nod detection and explored head shake detection.

Deep Multitask Gaze Estimation. We proposed a deep multitask framework for gaze estimation, with the following contributions. i) we designed a multitask framework which relies on both synthetic data and real data for end-to-end training. During training, each dataset provides the label of only one task but the two tasks are combined in a constrained way. ii) we introduce a Constrained Landmark-Gaze Model (CLGM) modelling the joint variation of eye landmark locations (including the iris center) and gaze directions. By relating explicitly visual information (landmarks) to the more abstract gaze values, we demonstrate that the estimator is more accurate and easier to learn. iii) by decomposing our deep network into a network inferring jointly the parameters of the CLGM model and the scale and translation parameters of eye regions on one hand, and a CLGM based decoder deterministically inferring landmark positions and gaze from these parameters and head pose on the other hand, our framework decouples gaze estimation from irrelevant geometric variations in the eye image (scale, translation), resulting in a more robust model. Thorough experiments on public datasets demonstrate that our method achieves competitive results, improving over state-of-the-art results in challenging free head pose gaze estimation tasks and on eye landmark localization (iris location) ones.

This work has been published in [Yu et al., 2018a].

Few-Shot User-Specific Gaze Adaptation. We addressed the problem of person-specific gaze model adaptation from only a few reference training samples. The main and novel idea is to improve gaze adaptation by generating additional training samples through the synthesis of gaze-redirected eye images from existing reference samples. In doing so, our contributions are threefold: (i) we design our gaze redirection framework from synthetic data, allowing us to benefit from aligned training sample pairs to predict accurate inverse mapping fields; (ii) we proposed a self-supervised approach for domain adaptation; (iii) we exploit the gaze redirection to improve the performance of person-specific gaze estimation. Extensive experiments on two public datasets demonstrate the validity of our gaze retargeting and gaze estimation

framework.

This work has been published in [Yu et al., 2019].

Besides the above contributions, I also participated in several other works. They are described below, but I do not plan to describe them in detail in this thesis.

Differential Gaze Estimation. To address the problem of person-specific bias in gaze estimation, a differential approach was proposed in this work where a network takes as input two eye images of the same person and learns to predict the gaze difference (person-specific bias eliminated by this way) between the two samples. When used for inference, some calibration samples are required for final gaze estimation. Experiments on 3 public datasets validate our approach which constantly outperforms state-of-the-art methods even when using only one calibration sample or when the latter methods are followed by subject specific gaze adaptation.

This work has been done with Gang Liu and has been published in [Liu et al., 2018, 2019].

Eye Movement Recognition. This work takes the temporal image frames as input and use the deep learning model to classify the gaze action into 3 categories, i.e. eye fixation, eye saccade and eye blink. Experiment on natural 4-party interactions demonstrates the efficacy of this approach.

This work has been done with Remy Siegfried and has been published in [Siegfried et al., 2019].

1.4 Thesis Origanization

Here we provide the organization of this thesis and briefly explain the content of each chapter.

Chapter 2. In this chapter, we first summarize the current head pose estimation approaches, including methods based on 2D image and methods based on RGBD data. We then discuss the literatures about head gesture recognition. Finally, the recent advances on gaze estimation approaches are presented.

Chapter 3. This chapter describes the proposed 3D head pose estimation methods, HeadFusion. We first give some brief background on 3D Morphable Models and 3D reconstruction. We then introduce how to combine these two elements to achieve an accurate and complete head representation for head pose tracking. Finally, experimental results on two public datasets are presented and discussed.

Chapter 4. This chapter presents how to extract head pose and body motion invariant dynamics from head pose data and apply them for head nod detection. Results on two datasets are

presented.

Chapter 5. In this chapter, we introduce the deep multitask framework for gaze estimation. Details about the Constrained Landmark-Gaze Model (CLGM) and our deep network framework which includes a network and a CLGM based decoder are presented. This chapter finally discusses experiments on both gaze estimation and eye landmark localization.

Chapter 6. In this chapter, we describe the approach of few-shot user-specific gaze adaptation. We cover elements like eye gaze redirection with synthetic images, domain adaptation with self-supervised training, gaze adaptation with gaze redirected samples. Finally, extensive experiments on two public datasets are demonstrated to validate our gaze retargeting and gaze estimation framework.

Chapter 7. In this chapter, we briefly summarize the works and contributions we made in this thesis. The limitations and perspectives of our proposed approaches are also discussed.

2 Related Work

In this Chapter we provide a literature review of works which are relevant to head nod detection and gaze estimation. As discussed in Chapter 1, head pose estimation is a requirement and preprocessing step for both head nod detection and gaze estimation. Therefore, literature review on head pose estimation will be discussed first in this chapter. Following the discussion on head pose estimation methods, we will summarize prior works on head nod detection. Finally, literatures on gaze estimation will be covered. When presenting and summarizing the related works, their contribution as well as their limitations will be discussed.

2.1 Head Pose Estimation

As discussed in Chapter 1 (Fig. 1.2), the head pose can be described by three rotation angles, pitch, yaw and roll, which correspond to the rotation around x, y and z-axis respectively. Therefore, the goal of vision based head pose estimation is to retrieve the pitch, yaw and roll angle from an observed head or face.

A number of literatures have been proposed to address the problem of head pose estimation. They can be divided into many categories. For instance, classification based method [Geng and Xia, 2014] which outputs discrete head pose categories and regression base method [Fanelli et al., 2011] which learns the mapping from head appearance to head poses (or from facial features to head poses [Cao et al., 2018]). In the following discussion, we mainly summarize the prior works according to the source data they deal with, i.e. 2D head pose estimation based on normal RGB images and 3D head pose estimation using RGB-D data obtained from depth sensors. Besides, we also discuss the literatures of 3D head representations which are fundamental to many model (the predefined head representation) based head pose estimation methods.

2.1.1 2D Head Pose Estimation

Due to the difficulty to model face appearance, early works on head pose estimation relied on keyframes, i.e. face image samples with associated head poses. The GAVAM model of Morency et al. [2008] is a typical example. It uses differential tracking to compare a given head image to observations in previous frames as well to a set of keyframes. It constantly updates the current keyframe pose estimates, and adds new ones when needed. Facial features tracking is an alternative line of work. Head pose estimation then becomes a secondary problem solved through PnP techniques. Constrained local models (CLM) [D.Cristinacce and T.F.Cootes, 2007] represent the appearance of local features as linear subspaces. Their location is found from filter responses of patch experts, constrained by a shape model. Later, Baltrusaitis et al. [2013] proposed the Constrained Local Neural Fields (CLNF), used in the OpenFace software, a variant of CLM addressing feature detection under more challenging scenarios. However, feature based methods suffer from self occlusions (encountered for large head poses), as they depend on features visibility.

Deep learning is gaining increased traction for tasks related to face analysis, such as detection [Li et al., 2015], verification [Taigman et al., 2014], and even gaze estimation from the full face [Zhang et al., 2016]. Deep learning has also been used for the localization of facial features [Sun et al., 2013; Zhou et al., 2013]. For instance, Sun et al. [2013] proposed a cascaded model composed of three levels, each of them having a set of parallel CNNs for which subgroups predict the location of the same landmark(s) and their response is averaged to reduce the variance. This process is repeated at the 3 levels, successfully achieving a coarse-to-fine prediction of the landmarks.

Besides detecting facial landmarks alone, Ranjan et al. [2016] proposed a multi-task learning framework where tasks of face detection, landmark localization, head pose estimation and gender recognition are achieved simultaneously. The multi-task network first extracts general features in the shallower layers, then the network is split into several branches and each branch deals with one task. One limitation of this work is that it requires a dataset which provides the labels of all the tasks. They later improved their work by leveraging tasks on multiple datasets [Ranjan et al., 2017], where additional tasks like smile detection, age estimation and face recognition is also incorporated. In both works, the facial landmark detection and the head pose estimation are modelled as two independent tasks thus their high correlation is not well utilized. Different from [Ranjan et al., 2016, 2017], Xu and Kakadiaris [2017] targeted only on head pose estimation and facial landmark detection. Similar to prevalent landmark detection methods, they used cascade regressors to adaptively learn the residual (residual error between prediction and ground truth) of landmark positions and head pose, and the head pose estimation and landmark detection tasks are jointly addressed by using global and local CNN features. Cao et al. [2018], however, adopted a two-step strategy where different types of facial keypoint features and keypoint relationships are first extracted with convolutional pose machines (CPMs) and then the head pose is regressed from these keypoints. Although obtaining head pose and the results of other tasks in a single framework can be efficient, how

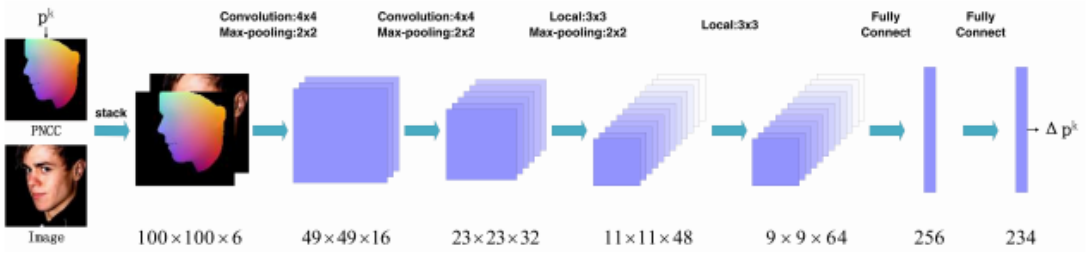


Figure 2.1: The inference diagram of [Zhu et al., 2016] at step K where the network takes the visual face and the shape from last step and learns to predict the residuals of model parameters.

to leverage multitasks and achieve high accuracy on all the tasks is still a difficult problem.

Instead of regressing head pose from visual appearance directly, some works have successfully combined deep learning with model based generative approaches. For instance, Zhu et al. [2016] (diagram shown in Fig. 2.1) utilized a 3D Morphable Model (3DMM) which incorporated both identity modelling and facial expression modelling. Starting from a visual face and a given shape of step K, the network regresses a residual parameter vector which corresponds to 7 variables, focal length, pitch, yaw, roll, translation, identity parameter and expression parameter. With the identity parameter and expression parameter, the authors relied on the 3DMM to reconstruct a 3D face supposed to have similar identity and expression as the input face. With the variable of pitch, yaw and roll, the 3D face is rotated to a similar head pose as the the input face. With the focal length and translation, the reconstructed and rotated 3D face model is projected to the image plane, and facial landmarks can be naturally extracted from the semantic face model. This paper also adopted a cascade strategy to learn the parameter residuals. A similar technique combining predefined model is reported in [Jourabloo and Liu, 2016]. Tewari et al. [2017], however, proposed an unsupervised approach to reconstruct a 3D face model from 2D appearance. The head pose and facial landmarks can also be obtained as byproducts. Compared with [Zhu et al., 2016; Jourabloo and Liu, 2016], they regressed from the input face a more diverse parameter vector which includes not only model parameters and pose parameters but also skin texture parameters and illumination parameters. With these parameters, the authors could build a textured 3D face. By projecting the 3D model to the 2D image plane, a loss measuring dense photometric alignment is employed to optimize the network.

2.1.2 3D Head Representation

Compared with classification methods and regression approaches, it is usually reported that model based methods can achieve better precision. Therefore, we review in this section the approaches to model 3D Head Representation which is a requirement for model based methods. We will not report on 2D head representations like ASM (Appearance Shape Model) [Cootes et al., 1995] and AAM (Active Appearance Model) [Cootes et al., 2001] since they often face

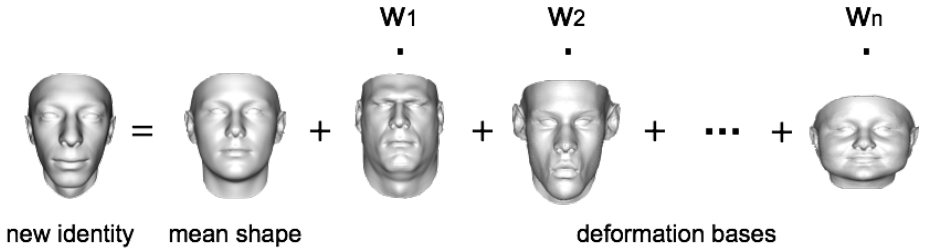


Figure 2.2: Identity modelling of 3D morphabel model.

challenges when dealing with large head poses.

Similar to 2D head representation like ASM and AAM, 3D Morphable Models (3DMM) have been proposed to model the 3D human face. The basic components include a mean face shape and a series of deformation bases which describe the face variations of different identities. By giving different coefficients to the deformation bases in a linear combination with the mean shape, the 3DMM can create 3D faces of various identities, as shown in Fig. 2.2.

The most commonly used public 3DMM is the Basel Face Model (BFM) [Paysan et al., 2009] which exploits PCA to learn the linear bases from 3D scans of 100 male and 100 female. However, since the data amount is relatively small and that most people scanned are Caucasians, it is difficult for BFM to achieve precise modelling for a broad set of people. To deal with this issue, a recent work [Booth et al., 2016] built a 3DMM from 9663 distinct identities which covers more ethnicities. Furthermore, they also established a collection of 3DMM models tailored by age, gender and ethnicity.

Instead of modelling variations using linear bases, Thomas and Taniguchi [2016] firstly used elastic registration with facial features to fit the head representation to a frontal posed head observation. Then a per-vertex adjustment map was computed and updated by computing the distance between the corresponding points in the head representation and the observation. However, the authors assumed the corresponding points were always located along the normal vectors of the head representation, which might be invalid if the initial head representation is too different from the observation.

Besides facial identity, the modelling of facial expressions is also a research focus. Cao et al. [2014] presented a public database of 3D facial expression blendshapes named FaceWarehouse. The facial expressions, then, can be modelled as a weighted sum of these blendshapes in a way similar to identity modelling. To model both the facial identity and the facial expression, recent works [Zhu et al., 2016; Jourabloo and Liu, 2016] adopted a simple strategy by linearly combining the term for identity modelling and the term for expression modelling together. However, the accuracy of modelling both parts through a linear combination remains a problem. Bouaziz et al. [2013] proposed a more promising method which personalized the expression blendshapes. The authors firstly obtained a deformation transfer operator through the identity modelling of neutral expression. Then the template expression blendshapes were

personalized using the deformation transfer operator. Similar blendshape personalization can also be found in [Thomas and Taniguchi, 2016]

2.1.3 3D Head Pose Estimation

Although visual based head pose estimation approaches have achieved important advances in recent years as mentioned in 2.1.1, they still suffers from many difficulties such as face appearance variability, extreme head poses, and illumination variations. The development of consumer 3D RGB-D sensors offers a better solution since the depth information measured from these sensors is inherently required for accurate 3D head pose estimation.

Indeed, the depth information excludes variabilities and uncertainties like skin color and illumination. It thus provides more robust features for 3D head pose estimation. For instance, Fanelli et al. [2011, 2013] proposed approaches relying on weak features extracted from depth patches to train a random forest regression model and acceptable performance of 4-5° error was reported. They also released a public dataset named BIWI for 3D head pose estimation. A more robust feature was designed by Papazov et al. [2015], more precisely a viewpoint invariant feature named triangular surface patch (TSP). It encodes the shape of the 3D surface of the face within a triangular area. With a simple nearest neighbor lookup, the depth map can be matched to a pose from a collection of predefined head models. This work reported a 2-3° error on BIWI dataset. Besides designing depth appearance features, Derkach et al. [2017] proposed an approach where 3D facial landmarks are first detected from depth patches. The 3D head pose is then regressed from the 3D positions of the detected landmarks. In their following work [Derkach et al., 2019], they applied tensor decomposition to the extracted features to model the 3D manifold resulted from the rotation angle. Their approach achieved high accuracy ($\sim 4°$ on BIWI) based on single depth frames.

Deep neural networks have also been used for this task. Different from the above approaches which extract features from depth maps, Venturelli et al. [2017] used convolutional neural network (CNN) to regress head pose from depth data directly. The interesting point is that the authors applied the idea of Siamese network to perform head pose estimation. Therefore, the optimization objective is not only the rotation angles of one observation, but also the difference of rotation angles between two observations. However, at test time, only one branch of the network is used. This technique, to some extent, handles person specific variations during training.

Despite the success of regression based methods, model based methods usually demonstrate superior performance, especially for RGB-D data. Weise et al. [2011] built a user specific 3D mesh face model offline using non-rigid registration, and then used the Iterative Closest Point (ICP) [Besl and McKay, 1992; Chen and Medioni, 1992] algorithm for real-time head tracking. Funes-Mora and Odobez [2012] extends [Amberg et al., 2008] by applying a multi-instance fitting to build an offline model and using an advanced point-to-plane ICP approach for tracking. However, as ICP is a local optimization technique, it requires a good initialization,

and thus often needs to process data at a high frame rate. To solve this problem, Meyer et al. [2015] combined ICP and Particle Swarm Optimization (PSO) together for joint tracking and online fitting, thus allowing to propose and evaluate multiple initializations. Higher pose estimation accuracy is achieved at the expense of a much higher computational cost.

Finally, some works have been proposed to model facial expressions which are non-rigid deformations. These works can be used to transfer the estimated expressions to animated avatars. Methods like [Bouaziz et al., 2013; Hsieh et al., 2015; Thomas and Taniguchi, 2016] model facial deformations through blendshapes which linearly extend a standard 3DMM. An advantage of these methods, as done by Bouaziz et al. [2013] is that by decomposing the face model, it is possible to retrieve the components related to face identity even under facial deformations, as well as adapting the facial deformation basis online. In a later work, the authors in [Hsieh et al., 2015] achieved robust head tracking under occlusion. They identified outliers by measuring the difference between the current observation and the head model set with the estimated head pose of previous frames. However, due to their focus, these papers lack a real evaluation of the head pose estimation component and tracking robustness in non near-frontal pose conditions.

2.2 Head Nod Detection

Early works proposed to detect head gestures using facial features. A.Kapoor and R.Picard [2001] present a technique to recognize head nods and shakes based on two Hidden Markov Models (HMMs) using 2D coordinate results from an eye tracker. In [Tan and Rong, 2003], an AdaBoost algorithm and anthropomorphic measures are applied to detect user's face and locate eye zone, respectively. Head movements are then derived from the eye location, and are then used within a discrete HMM to detect head nods and shakes.

Recent works, however, have been developed based on 3D head trackers. The approach in [Nakamura et al., 2013] models a nod as a velocity pattern of the pitch angle. The pattern is extracted by detecting whether the 3D head tracker changes from a point below a negative threshold to a point above a positive threshold within a certain time interval. The authors in [Wei et al., 2013] describes another method in which 5 head states (up, down, left, right and still) are distinguished. Then the head nod and shake are further recognized with two HMMs.

These approaches show good performance in simple scenarios where listeners use exaggerated head gesture to answer yes or no. But their performance drops significantly in detecting nods in natural face-to-face conversations where nods are more subtle and less explicit, because these methods tend to define nods as a sequence of head positions, which is a noisy feature to extract.

To better characterize and exploit the nod oscillating nature, other approaches use frequency features from the Fourier transform applied to head velocities. For instance, Morency et al. [2007] use them as well as contextual features like lexical information or prosodic cues from an

embodied conversational agent (ECA) to predict head nods of human, in a scenario involving a human interacting with an ECA. Nguyen et al. [2012] develop a multimodal method using frequency feature and audio based self-context by taking into account the influence of the speaking status of people on the dynamics of the head gestures. In this approach, the head velocities are computed at three arbitrarily defined points in a bounding box of a face tracker whose instantaneous motion field is estimated, using a robust and multi-resolution optical flow computation method [Odobez and Bouthemy, 1995]. The authors apply a Fourier transform with Gaussian temporal window to these velocities. Fourier features are then used to train two separate classifiers, one for speakers and the other one for listeners. Compared to [Nakamura et al., 2013; Wei et al., 2013; A.Kapoor and R.Picard, 2001; Tan and Rong, 2003], frequency features result in a better description of fine head movements. These two approaches have shown good performance in the context of human-computer interaction (HCI) [Morency et al., 2007] and natural conversation [Nguyen et al., 2012]. The features characterizing the context, in which nods occur, have also been proved useful for improving the detection.

A main limitation of all these methods is the constraint linked to the position of the camera. They assume that in the training and test video, interlocutors have a similar head pose, and very often a frontal one. Therefore, these approaches cannot achieve pose invariance when camera position changes, or when people faces are oriented in more variable direction, e.g. when observing people in multiparty situations.

2.3 Gaze Estimation

As mention in Chapter 1, the problem of gaze estimation can be categorized into 3 classes, Gaze Fixation Point Estimation, Gaze Following and 3D Gaze Estimation. Although this thesis mainly targets at 3D Gaze Estimation which attempts to infer the 3D Line of Sight of the eye, we also investigate and discuss the other two types of gaze estimation since all of them are closely related. At the end of this section, we also summarize some popular public datasets for remote gaze estimation.

2.3.1 Gaze Fixation Point Estimation

Gaze Fixation Point Estimation aims to predict the 2D fixation point of human gaze on a flat surface. One of its popular applications is to predict human gaze on screens of mobile devices to enable Human Computer Interaction.

Krafka et al. [2016] proposed to learn the gaze fixation point on smart phone screens using images taken from the front-facing cameras of iPhones. Their network takes 4 image channels as inputs, including one face image, two eye patches and one face position map, shown in Fig. 2.3. To train their network, they collected a large dataset with 2.5M frames and an accurate estimator with 2cm error is achieved. A similar dataset is proposed in [Huang et al., 2017] where the scenario corresponds to an interaction between people and an iPad. Different

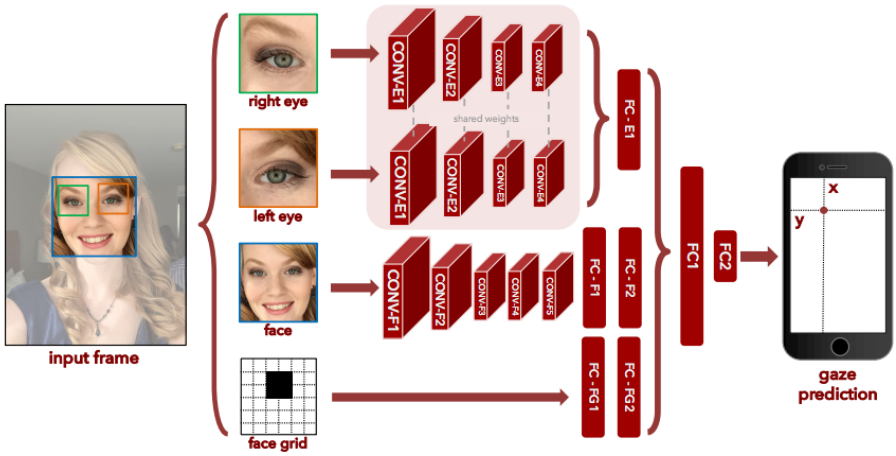


Figure 2.3: Framework of iTracker [Krafka et al., 2016].

from [Krafka et al., 2016] which relied on deep networks, this work used some traditional features like HoG and LBP to regress gaze. In a recent work, Zhang et al. [2018] trained a cross-device gaze model which consists of shared eye image feature extraction layers and device specific branches. They collected a new dataset covering five common devices and demonstrated that it is possible to achieve person specific gaze estimators on multiple devices.

The size of gaze fixation point dataset is usually larger than that of 3D gaze direction dataset, since it is much easier to annotate 2D fixation point. However, it is hard to apply the gaze fixation point estimator to a broad range of devices or scenarios, thus the application scope is somehow limited.

2.3.2 Gaze Following

The gaze following problem refers to the inference of the objects people are looking at. Different from gaze fixation point estimation or 3D gaze estimation whose usual inputs are human faces or human eyes, the images gaze following deals with can cover a wider range and the people in the images are less constrained. Besides, the gaze object and the person who is looking at it are usually required to appear in the same image.

To our best knowledge, Recasens et al. [2015] proposed the first work on gaze following. In this work, the network consists of two pathways, the saliency pathway which takes the original image as input and spots the salient regions in the image, and the gaze pathway which takes the cropped head image and the location map of the head within the image, and predicts the candidate regions people might look at. The output heatmaps of the two pathways are merged with element-wise product and fully connected layers are used to get the final gaze position, shown in Fig. 2.4. Continente et al. [2017] later extended their work to video streams where cross-frame gaze following (gazing people and gazing object in different frames) was addressed by learning the geometric relationship between different views.

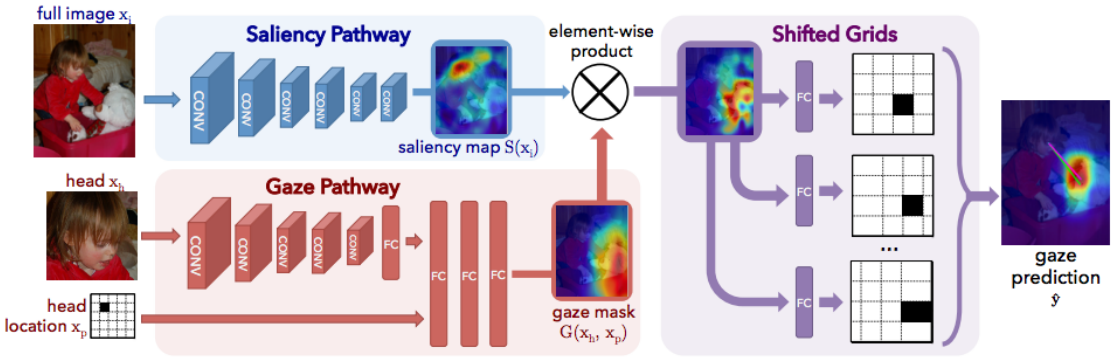


Figure 2.4: Network architecture of [Recasens et al., 2015] which consists of a saliency pathway and a gaze pathway.

The limitations of gaze following approaches are mainly two-folds: i) the gaze following models might learn to predict the human head pose rather than the human gaze; ii) it is difficult for gaze following models to handle scenarios where people are looking to the front (or near front) and the gaze object is not within the image.

2.3.3 3D Gaze Estimation

The 3D gaze estimation task which attempts at retrieving the 3D line of sight of eyes is the main focus of this thesis and we mainly investigate the vision based non-invasive and non-active (i.e. without infra-red sources) methods which have addressed it.

Traditional Gaze Estimation Methods

Traditional 3D gaze estimation can be grouped into two categories, the geometric based methods (GBM) and appearance based methods (ABM) [Hansen and Ji, 2010]. GBM rely on a geometric model of the eye whose parameters (like eye ball center and radius, pupil center and radius) can be estimated from features extracted in training images [Venkateswarlu et al., 2003; Wood et al., 2016a; Ishikawa et al., 2004; Wood and Bulling, 2014; Gou et al., 2016, 2017; Timm and Barth, 2011; Villanueva et al., 2013] and can further be used to infer the gaze direction. They usually require high resolution eye images from near frontal head poses to obtain stable and accurate feature detection, which limits the user mobility and their application to many of our settings of interest. By learning a mapping from the eye appearance to the gaze, ABM [Tan et al., 2002; Huang et al., 2017] are more robust to lower resolution images. They usually extract visual features like Retinex feature [Noris et al., 2011] and mHOG [Martinez et al., 2012], and train regression models such as Random Forest [Sugano et al., 2014], Adaptive Linear Rregression [Lu et al., 2011], Support Vector Regression [Martinez et al., 2012] for gaze estimation. Very often ABM assumed a close-to-static head pose, but recently head pose

dependent image normalization have been tested with success, as done for instance in [Funes-Mora and Odobez, 2016] where a 3D morphable model is used for pose normalization. In general, ABM require a large amount of data for training and they do not model the person gaze variation explicitly with a model.

Gaze Estimation with Deep Learning

Recent works started to use deep learning to regress 3D gaze directly. Zhang et al. [2015] proposed a shallow network for gaze estimation. Different from [Funes-Mora and Odobez, 2016], this paper learns to predict head pose dependent gaze. To achieve this goal, the network concatenated the head pose information with the features learned from the input eye when regressing gaze with fully connected layers. Besides, this work also introduced a dataset named MPIIGaze for 3D gaze estimation. While it provides challenging data, the images were collected with different laptops and over the time length of 3 months. The participants were mainly facing the laptops so most of the faces are close to frontal head pose. The number of participants is also small, which may limit model generalization when being used for training. In their following work, Zhang et al. [2017] used a VGG network as the backbone and trained a much deeper network for gaze estimation. Significant performance improvement was achieved by this network.

Researchers soon realized that it is not enough to predict gaze from a single eye. Cheng et al. [2018] proposed to use both eyes for gaze estimation. In their framework, an asymmetric regression network was used to merge the information from both eyes for gaze prediction. An evaluation network then was applied to adjust the regression strategy by evaluating the gaze estimation performances of the two eyes. Some works, however, took a step further by processing the full face. For instance, Zhang et al. [2016] fed the full face to a network which used the attention mechanism to spot regions with high correlation to gaze. In this work, the head pose information was not explicitly used or estimated. Zhu and Deng [2017], however, proposed a framework which achieved both head pose estimation and gaze estimation. Concretely, their network consists of two pathways where one processes the full face to get the head pose while the other deals with the eye image to estimate the gaze in the head coordinate system. The gaze, then, is geometrically transformed to world coordinate system through the head pose, as was done in the framework of [Funes-Mora and Odobez, 2016]. Although the geometrical modelling of head pose and gaze makes sense, it is hard to compare the performance of their methods with other approaches since the dataset they use is not public. Having noticed that traditional convolution filters might not be suitable for delicate tasks like gaze estimation, Chen and Shi [2019a] adopted dilated convolutions which can efficiently increase the size of receptive field without reducing the spatial resolution to extract high level features of eyes. By processing the full face as well as the two eyes, some improvement is achieved by their architecture.

All deep learning based methods rely heavily on the available data. However, as mentioned in Chapter 1, annotating 3D gaze is complex and expensive. Therefore, most current public

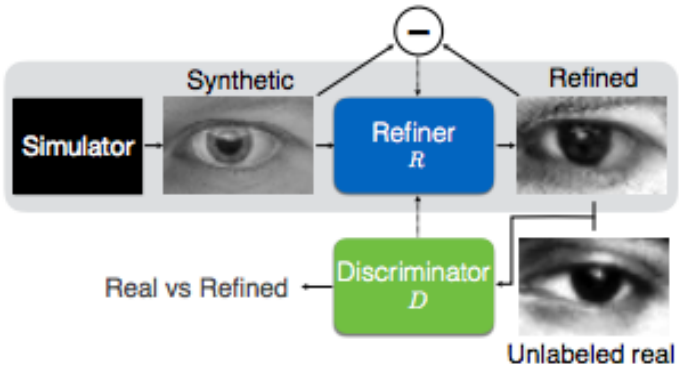


Figure 2.5: Framework of SimGAN [Shrivastava et al., 2017].

3D gaze datasets are relatively small (in both samples and participants) compared with other computer vision tasks like image recognition and object detection, which makes it difficult to train an accurate and robust gaze model.

Gaze Estimation with Synthetic Data

To address the data lacking issue, Wood et al. [2016b, 2015] attempted to use computer graphic techniques to generate synthetic eyes for the training of gaze estimator. Although the estimator trained only from the synthetic images did work on real eye images, its performance was still significantly lower than those trained on real eye images. The domain gap could be the main cause here and the development of adversarial learning offers a possible solution to domain adaptation. Shrivastava et al. [2017] proposed SimGAN, a framework which uses conditional Generative Adversarial Network (GAN) [Goodfellow et al., 2014] to refine the synthetic eye images to a target domain, real eyes, shown in Fig. 2.5. On one hand, the GAN based approach attempted to make the appearance of synthetic eyes as close as possible to real eyes. On the other hand, the network also attempted to preserve some key information like gaze and identity during the refinement. By adapting the synthetic data with SimGAN, the gaze error could be reduced by 3°. Besides domain adaptation, GAN has also been used for eye image generation directly [Wang et al., 2018] where the eye generation starts with a user define gaze and a corresponding eye shape generated by a graphical model. Then the eye appearance is rendered using a conditional GAN. The performance of the gaze estimation trained with the generated eye samples is close to SimGAN.

Despite the progress being made by these computer graphics based or network based eye generation approaches, there is still some domain gap between the appearances of synthetic data and real data. Furthermore, as pointed in [Yu et al., 2018a], for synthetic data, there exists a between-dataset systematic bias regarding the gaze ground truth. A better example of synthetic eye generation is given by [Sugano et al., 2014] where real eye samples of multiple viewpoints were collected to reconstruct the 3D shape of the eyes. Then dense multi-view eye samples were generated using the reconstructed eye models. Despite the good quality and

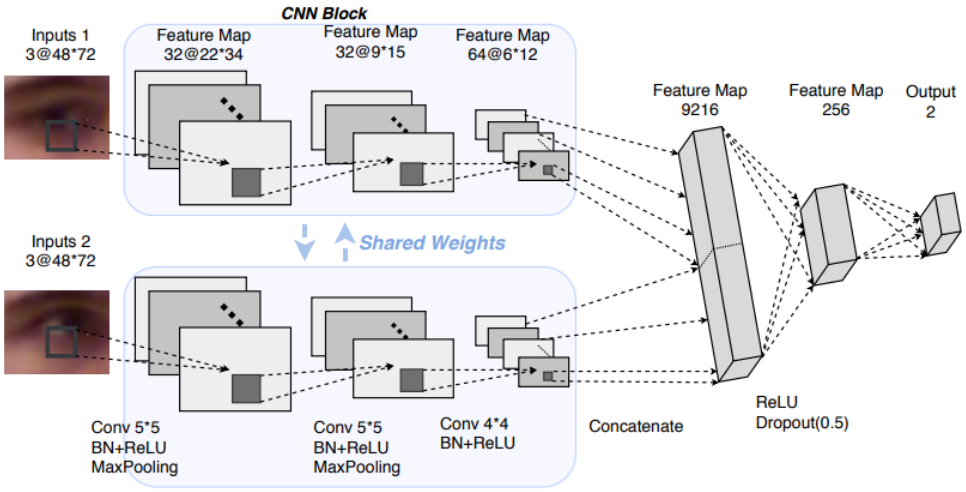


Figure 2.6: Differential gaze estimation [Liu et al., 2018].

reliable ground truth of the generated samples, the setup of the whole pipeline is complex and expensive. Furthermore, the synthetic samples mainly differed in head pose, but the gaze in head coordinate system was not diverse enough.

Person-Specific Gaze Estimation

As pointed out in Chapter 1, gaze can not be fully estimated from the visual appearance since the alignment difference between the optical axis (the line connecting the eyeball center and the pupil center) and the visual axis (the line connecting the fovea and the nodal point [Funes Mora and Odobez, 2014]) is person specific, and vary within -2 to 2 degrees across the population. Therefore, applying a single model to various persons can be inaccurate. To address the personal bias issue, Liu et al. [2018] proposed a differential framework which learns to predict the gaze difference of two person-specific samples, shown in Fig. 2.6. It is assumed that the personal bias of gaze difference is much smaller than that of gaze. When used for testing, the model requires some annotated reference samples for gaze inference. But even with one calibration sample, the approach is still better than the baseline methods (direct regression). The experimental results demonstrated significant improvement brought by this calibration approach. Following the linear calibration approach also proposed in [Liu et al., 2018, 2019], Chen and Shi [2019b] showed that by adopting a much deeper network, the personal bias can be corrected with a single gaze point, which makes the calibration procedure much easier. Another way to achieve person-specific gaze estimation is to fine-tune the network using specific samples of a user directly [Yu et al., 2019] where it is shown that such a strategy can lead to a large improvement. Later, Park et al. [2019] addressed the person-specific adaptation problem in a more principal way. They treated the learning of person-specific estimator as learning different tasks and relied on Model-Agnostic Meta-Learning (MAML) to achieve few shot sample adaptation.

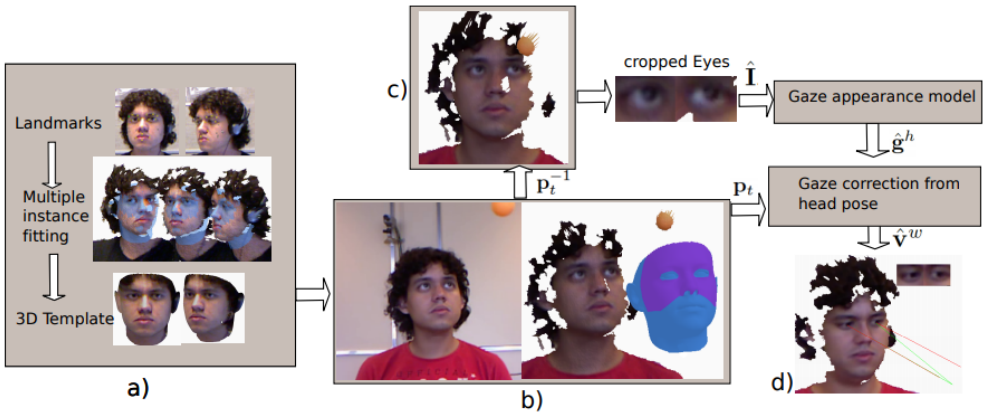


Figure 2.7: Eye rectification with head pose [Funes-Mora and Odobez, 2012]

The above methods basically used person-specific adaptation to address the issue of person-specific bias. The limitation is that they require person-specific samples, which is not convenient in real application. In some recent works, the person-specific bias is explicitly modelled when training a generic gaze estimator. For instance, Lindén et al. [2018] proposed to model the personal bias as calibration parameters which is supposed to be optimized during training. When used for testing, user can either collect person-specific samples to learn the calibration parameters or simply set the calibration parameters to their mean values. A similar strategy can be found in [Xiong et al., 2019] where the Mixed Effects Neural Networks (MeNets) models the fixed effect and random effect separately. However, to obtain accurate person-independent gaze models, these two approaches might require datasets with a large number of participants.

Gaze Estimation with Head Pose Information

In gaze estimation, how to use the head pose information is an important problem since the perceived gaze is not only determined by the eye movement by also by the head pose w.r.t the camera. In this section, we summarize 5 approaches to integrate head pose in gaze estimation.

Eye rectification with head pose. This approach [Funes-Mora and Odobez, 2012, 2016] (Fig. 2.7) usually requires RGB-D data. It first estimates the head pose from the observed face or head, then it projects the eye patches to the frontal head pose space by performing a rigid geometric transformation (derived from the head pose) to the point clound data and the texture data. The training and testing of gaze models thus are conducted on eye images of the same head pose, which results in precise performance. Although high accuracy can be achieved, this approach relies on RGB-D sensors, which might limit its application scope. Besides, the rectified eye patches can be distorted if the head pose estimation is not accurate enough.

Geometric transformation within gaze estimator. Similar to [Funes-Mora and Odobez, 2012, 2016] somehow, Zhu and Deng [2017] (Fig. 2.8) learns to predict the gaze in head coordinate

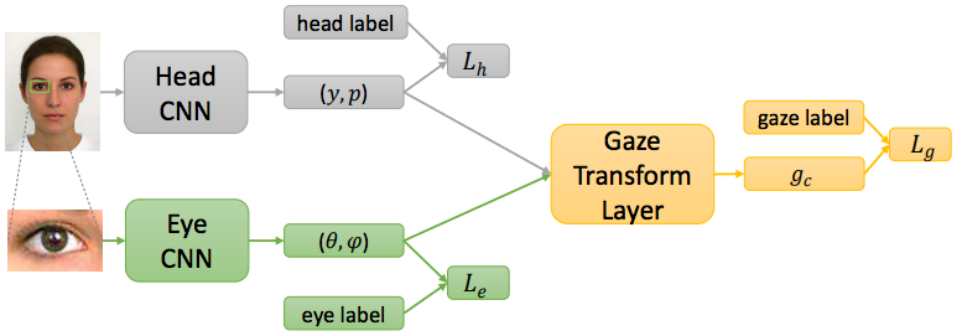


Figure 2.8: Geometric transformation within gaze estimator [Zhu and Deng, 2017]

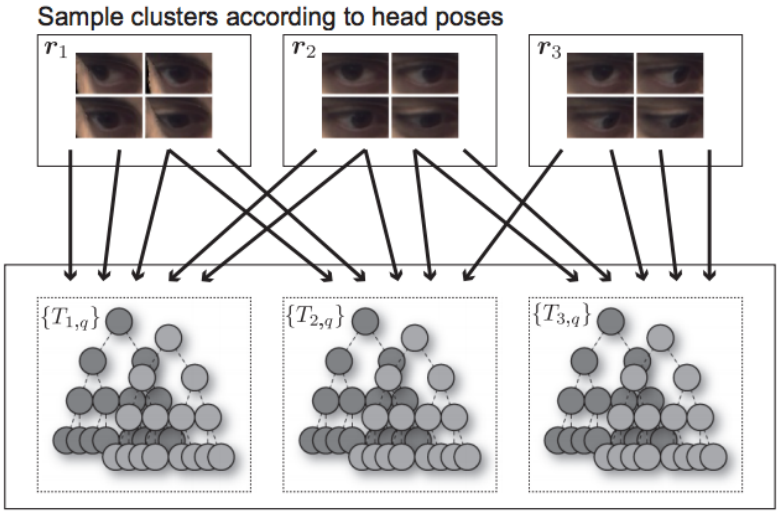


Figure 2.9: Gaze estimation based on head pose clusters [Sugano et al., 2014]

system. But they later convert the gaze to world coordinate system through a geometric transformation layer and the network is trained in an end-to-end fashion. Although some improvement is achieved, it can be difficult to learn accurate gaze in head coordinate system from eye images of various head poses.

Gaze estimation based on head pose clusters. Sugano et al. [2014] proposed to learn a series of gaze models, each of which targets at eye images of a specific head pose cluster, shown in Fig. 2.9. Although the motivation is reasonable, learning multiple gaze models can be time and computational expensive.

Feature concatenation. Zhang et al. [2015] is a typical example of this approach where a new feature is constructed by concatenating the eye features with the head pose. The network then processes the new feature using the fully connected layers to compute the final gaze w.r.t the camera, shown in Fig. 2.10. But it might not be accurate enough to model the geometric relationship with nonlinear projections.

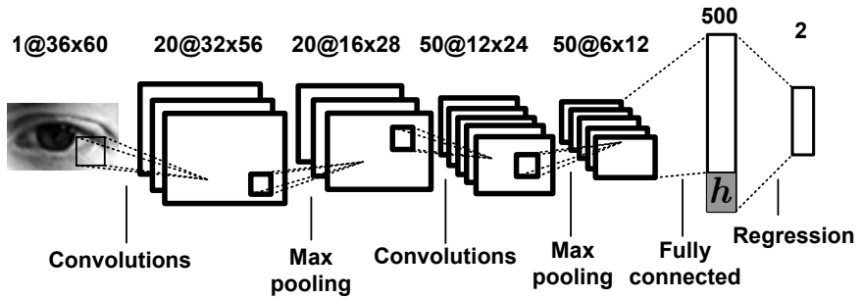


Figure 2.10: Feature concatenation [Zhang et al., 2015].

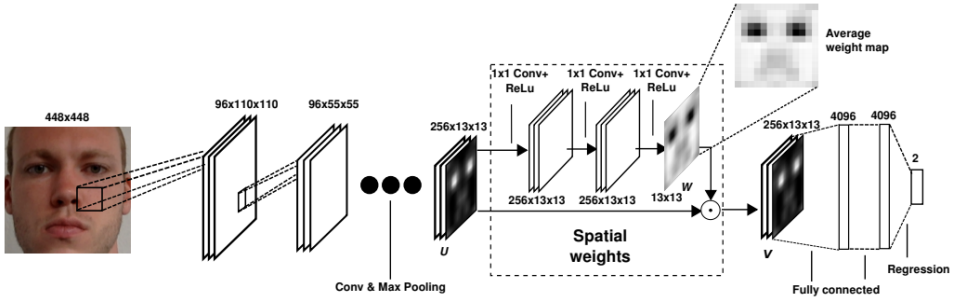


Figure 2.11: Implicit modelling of head pose [Zhang et al., 2016].

Implicit modelling of head pose. This approach [Zhang et al., 2016] (Fig. 2.11) usually requires full face images and it learns to predict the gaze w.r.t the camera directly without taking the head pose information into account explicitly. However, this approach might be challenged by scenarios where large head poses are presented since the label information provided is limited.

2.3.4 Gaze Estimation Datasets

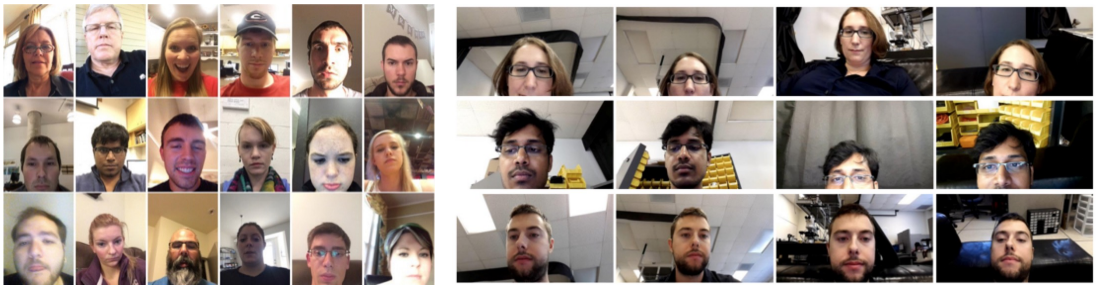
There have been a number of datasets proposed for gaze estimation. In this section, we summarize and discuss the property and setup of these datasets. Some properties of the datasets have been displayed in Table. 2.1.

GazeCapture

GazeCapture [Krafka et al., 2016] was collected aiming at gaze fixation point estimation. In data collection, people were required to look at the dots appeared on the screen of their mobile devices (iPhone or iPad in GazeCapture) and an APP would take the pictures of the participants' faces. Some sample images of GazeCapture can be found in Fig. 2.12(a). This dataset collected ~2.1M images from ~1.5K subjects. It also provides the cropping information

Table 2.1: Some properties of the datasets

Task	Dataset	Full Face	# Subject	# Total	Gaze Pitch	Gaze Yaw
Gaze Fixation Point Estimation	GazeCapture	✓	~1.5K	~2.1M images	-20°~20°	-20°~20°
	TabletGaze	✓	51	816 videos	-15°~0°	-20°~10°
Gaze Following	GazeFollow	✓	130K	122K images	/	/
	VideoGaze	✓	/	224K images	/	/
3D Gaze Estimation	MPIIGaze	×	15	213K images	-5°~20°	-40°~20°
	MPIIFaceGaze	✓	15	213K images	-5°~20°	-40°~20°
	UTMultiview	×	50	~1M images	-55°~65°	-80°~80°
	ColumbiaGaze	✓	56	5.88K images	-10°~10°	-45°~45°
	Eyediap	✓	16	94 videos	-45°~45°	-45°~45°
	RT-GENE	✓	15	122K	-30°~30°	-40°~40°
	SynthesEyes	✓	10	11K	-50°~50°	-50°~50°
	UnityEyes	✓	/	user defined	user defined	user defined



(a) GazeCapture samples (b) TabletGaze samples

Figure 2.12: Samples of gaze fixation point datasets, all taken by the frontal cameras of mobile devices. (a) GazeCapture samples. (b) TabletGaze samples.

of faces and eyes.

TabletGaze

TabletGaze [Huang et al., 2017] was collected under a similar setup as GazeCapture. The main difference includes: i) the mobile device for data collection was restricted to a Samsung Galaxy Tablet; ii) the participants were required to perform 4 body postures in data collection, with more constraint compared with GazeCapture; iii) the data was recorded as videos; iv) there are much less participants (51) in TabletGaze. Some sample images are displayed in Fig. 2.12(b).

GazeFollow

GazeFollow [Recasens et al., 2015] was collected targeting at the problem of gaze following. It merged several major datasets containing people. Annotators used online tools to mark persons' eyes and the objects they believed people were looking at. Note that the gaze annotations can be multimodal in the dataset since different annotators can mark different gaze

(a) GazeFollow samples (b) VideoGaze samples

Figure 2.13: Samples of gaze following datasets. (a) GazeFollow samples where the end points of the rays from human heads denote the multimodal gaze objects. (b) VideoGaze samples. The orange bounding boxes in the first column denote the gazing persons while the rest columns on the right are the candidate frames where a red bounding box means no gaze object in the frame while a green dot in the green bounded frame denotes the gaze object.

objects when annotating the same image. Some sample images and multimodal annotations (denoted by the end points of the rays from human heads) are shown in Fig. 2.13(a).

VideoGaze

VideoGaze [Continente et al., 2017] is a video version of GazeFollow. As the main characteristic, VideoGaze provides cross-frame gaze annotation, which means the person and the object the person was looking at can appear in two different frames. Some samples can be found in Fig. 2.13(b) where the orange bounding boxes in the first column denote the gazing persons while the green dots in the right columns denote the gaze object people were looking at.

ColumbiaGaze

ColumbiaGaze [Smith et al., 2013] has a similar collection setup (chin rest to stabilize head pose and multiple cameras for multi-view samples) as UTMultiview. It collected 5.88K images from 56 subjects with varying gazes and head poses. As a main difference to other datasets, the gazes of ColumbiaGaze are sparse and discrete distributed. Fig. 2.14(c) demonstrates the setup and example images from ColumbiaGaze.

UTMultiview

UTMultiview [Sugano et al., 2014] collected 64K images from 50 subjects. During data collection, the participants were asked to look at the targets on the screen and a chin rest was used to fix the head poses of the participants. The images were taken in synchrony by 8 cameras to create multi-view samples. Afterwards, to further expand the data size, the multi-view samples were used to reconstruct the 3D shapes of the eyes which were projected to 2D image plane to generate dense synthetic samples. The setup and sample images are shown in Fig. 2.14(b).

Eyediap

Eyediap [Funes Mora et al., 2014] consists of 94 sessions with different participants (16 participants), recording conditions (2 conditions with different days, illumination and distance to camera), visual targets (continuous screen target, discrete screen target and 3D floating target) and head poses (static head pose and moving head pose). The main characteristic of this dataset is that it used Kinect sensor to collect data. With depth information from Kinect sensor, the 3D gaze annotation, the head pose estimation and eye position detection become much easier. It is even possible to rectify the eyes to a frontal head pose. Besides the normal RGB videos and the corresponding depth sequences, this dataset also recorded HD videos. The collection setup and sample images can be found in Fig. 2.14(d).

MPIIGaze

MPIIGaze [Zhang et al., 2015] was collected for 3D gaze estimation. The data was collected while people were using their laptops in their daily work with an application appearing to require them to look at some calibration points. The collection duration covers over 3 months. Then the data was post-processed as follows. First, the eyes in the images were detected with a face model consisting of 6 landmarks. Second, for the purpose of 3D gaze estimation, the face images, eye locations and the 2D gaze fixation points on the screen were projected to a normalized camera space. Finally, the normalized eyes were cropped from the images and the 3D gaze was computed within the normalized space. This dataset contains 213K images from 15 participants, and it does not provide full face images. The collection setup and some sample images are demonstrated in Fig. 2.14(a).

MPIIFaceGaze

MPIIFaceGaze [Zhang et al., 2016] was an extension of MPIIGaze. It provides the full faces of the participants which were not included in MPIIGaze. The positions of the 6 facial landmarks used to compute the head pose are also provided.

RT-GENE

RT-GENE [Fischer et al., 2018] is a recent dataset. Different from all the datasets above, it used an eye tracking glass to annotate gaze and a motion capture system to track head pose. To make the collected image more natural, the authors used GAN to remove the eye tracking glass from the face images. Some sample images are shown in Fig. 2.14(e).

SynthesEyes and UnityEyes

SynthesEyes [Wood et al., 2015] and UnityEyes [Wood et al., 2016b] are two synthetic datasets for 3D gaze estimation. Their difference includes: i) SynthesEyes defined 10 head models for

synthetic eye generation while UnityEyes generate synthetic eyes with random textured skin, sclera, iris and pupil. ii) SynthesEyes provides limited number of samples and gaze variations while the number of samples and gaze variations of UnityEyes is user defined. Synthetic samples of the two datasets can be found in Fig. 2.15. In this thesis, when we use synthetic dataset for gaze estimation, we only select UnityEyes because of its diversity.

In our experiment of 3D gaze estimation, we selected datasets of UTMultiview, Eyediap, ColumbiaGaze and MPIIGaze for evaluation. The former two datasets are used in Chapter 5 since they provide relatively large amount of data and precise head pose annotations which are necessary for our proposed multitask approach. Chapter 6 selected ColumbiaGaze dataset since this dataset has the most participants. Then we also used MPIIGaze in this chapter because we would like to test whether our gaze redirection approach works on noisy images.

2.4 Conclusion

In this section, we discuss and summarize the main characteristics and limitations of the previous works. We will also briefly introduce how our contributions can address these limitations.

Head Pose Estimation.

By investigating the literatures on head pose estimation, we found that the model based approaches relying on depth information usually show superior performance to other methods. A typical model they use is the 3D Morphable Model (3DMM) which provides a mean face shape and a series of deformation bases and the error achieved can be as low as $\sim 2^\circ$ (on BIWI dataset). It is difficult and unnecessary to further reduce the error since the ground truth of head pose itself is estimated by model based registration. Therefore, compared with the low error, we believe it is more important to achieve robust and uninterrupted head pose tracking. However, challenges are facing 3DMM based approaches when extreme head poses are presented because the 3DMM usually only covers the face region and lacks the support for the side and up part of the head.

In this thesis, we propose an approach which combines the strengths of a 3DMM model fitted online with a prior-free reconstruction of a 3D full head model providing support for pose estimation from any viewpoint. With such a complete head representation, the head pose tracking can be more accurate and robust in challenging scenarios of extreme head pose.

Head Nod Detection.

Section. 2.2 points out that the previous works on head nod detection are constrained to the position of the camera. They assumed a near-frontal head pose in the videos and can not generalize well to data with different head poses.

In this thesis, we compute the head motion dynamics within the head coordinate system

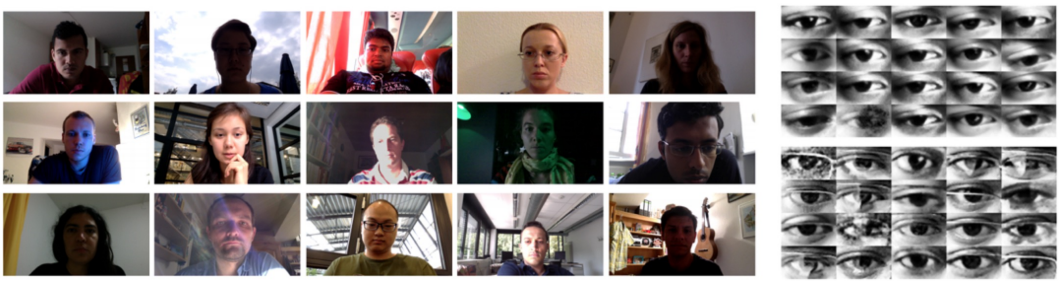
instead of the world coordinate system, resulting in camera pose invariant motion features. A nod detector with such features can generalize well to videos taken from a broad range of viewpoints.

Gaze Estimation.

Section. 2.3.3 made a thorough discussion on the 3D gaze estimation approaches and Section. 2.3.4 listed some public datasets on 3D gaze estimation. The limitations and challenges of 3D gaze estimation approaches or dataset have been summarized in Section. 1.2.3 (Lack of Data, Systematic Bias, Eye Cropping, Person-Specific Bias).

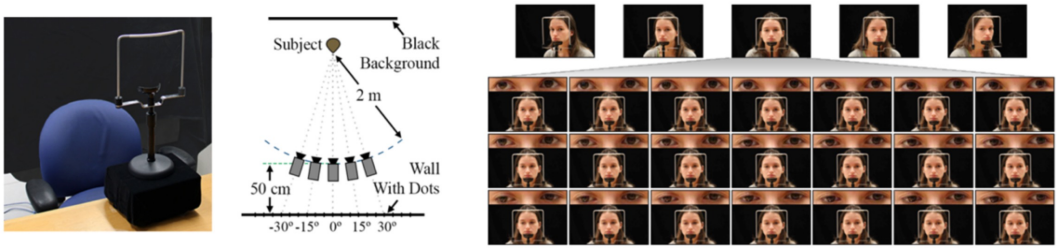
To address these challenges, we first proposed a multitask learning approach which relies on the synthetic dataset (UnityEyes). The synthetic data not only increases the amount of training data but also provides the label of another task (eye landmark detection), thus addressing the data lacking problem to some extent. Besides, we also introduced a Constrained Landmark-Gaze Model (CLGM) modelling the joint variation of eye landmark locations (including the iris center) and gaze directions. The CLGM allows to minimize the systematic bias between datasets through a linear adaptation model and overcomes the eye cropping problem by detecting eye landmarks.

We then address the problem of person-specific bias by proposing a few-shot gaze adaptation approach. The target is to adapt a generic gaze model to a new user with only a few reference samples. Our main idea is to generate more person-specific samples through the gaze redirection technique. We then finetune the generic gaze model with the expanded reference samples. Compared with generic gaze models, a personalized gaze estimator can have more accurate performance.

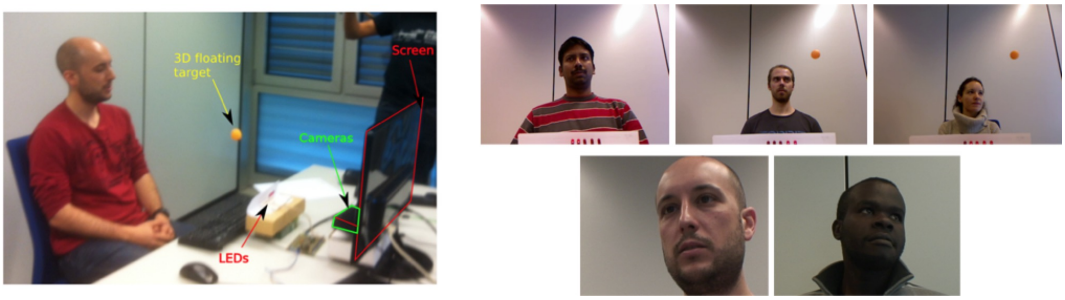


(a) MPIIGaze dataset, collection setup (left) and samples (right)

(b) UTMultiview dataset, collection setup (left) and samples (right)



(c) ColumbiaGaze dataset, collection setup (left) and samples (right)



(d) Eyediap dataset, collection setup (left) and samples (right)

(e) RT-GENE dataset, collection setup (left) and samples (right, including images with and without eye tracking glasses)

Figure 2.14: 3D gaze estimation datasets, collection setups and samples. (a) MPIIGaze. (b) UTMultiview. (c) ColumbiaGaze. (d) Eyediap. (e) RT-GENE.

(a) SynthesEyes samples

(b) UnityEyes samples

Figure 2.15: Samples of synthetic datasets. (a) SynthesEyes samples. (b) UnityEyes samples.

3 3D Head Pose Estimation

In this chapter, we describe in detail the 3D head pose estimation method (HeadFusion) used in this thesis. Our 3D head pose estimation is based on RGB-D data since the depth information is inherently required for 3D head pose estimation. Our method belongs to model based approaches and relies on a 3D head representation which combines the benefits of 3D Morphable Model (3DMM) and online 3D head reconstruction.

The rest of this chapter is organized as follows: we motivate our main idea and list the contributions in Chapter. 3.1; A brief background on 3D reconstruction is given in Chapter. 3.2; Chapter. 3.3 makes an overview on our approach; The methods on head pose estimation and head representation modelling are introduced in Chapter. 3.4 and Chapter. 3.5 respectively; We present our collected dataset UbiPose and the experiment protocol in Chapter. 3.6; The results are reported and analyzed in Chapter. 3.7; Finally, we make a summary of our approach in Chapter. 3.8.

3.1 Motivation and Contributions

As mentioned in Chapter 2, when using RGB-D data, model based approaches [Meyer et al., 2015] which rely on a predefined face model and retrieve the head pose parameter by minimizing the discrepancy between the observation and the head model usually report more precise results than regression based methods [Fanelli et al., 2011; Papazov et al., 2015]. They often use 3D Morphable Models (3DMM) [Blanz and Vetter, 1999] to retrieve the subject's face model, since they provide a linear and low dimensional representation of the 3D facial shape variations across a population allowing online and well constrained model adaptation by finding the coefficients for the subject of interest. Furthermore, the 3DMM model also provides semantic information which is fundamental for other tasks such as gaze estimation. However, as illustrated in Fig. 3.1a, the limitation of most 3DMM models is that they only cover the frontal region of the face. The top, side and back parts of the head are missing since it is actually quite difficult to extract a linear statistical basis from the large variations of the hair (and even the ears) in these parts across the population.

(a) BFM model

(b) online reconstruction

(c) estimation results

Figure 3.1: Head model and pose estimation. (a) the 3DMM head representation only covers part of the head. (b) online head reconstruction progressively incorporating observations. (c) head pose estimation using only a 3DMM (top) and incorporating a reconstruction component (bottom).

Most applications so far consider applications where the subject's face is nearly frontal. However, there are many applications where such an assumption can not be guaranteed, like when setting sensors in the environment and monitoring people's activities and interactions. Being able to track the head uninterruptedly in non-constrained natural scenarios (such as our UBImpressed sequences, shown in Fig. 3.6c), where unexpected cases such as fast motions, occlusions, and more profile or adverse poses are presented, is thus also very important. However, a model only relying on the frontal face representation lacks the support to handle these cases, as shown in the top line of Fig. 3.1c.

In this chapter, we thus propose a novel robust and accurate head pose estimation method which fuses the strengths of two head representations:

- a 3DMM facial model automatically adapted online from a collection of samples, able to provide very accurate head pose estimations for near frontal head poses, but which has difficulties at tracking heads otherwise;
- an online reconstruction 3D head model based on a variant of KinectFusion [Newcombe et al., 2011], bringing the robustness of tracking of the head over a 360°range.

Furthermore, we propose to exploit a symmetric regularizer for the non-linear fitting of the 3DMM, preventing unwanted deformations that can degrade performance when mainly observing the face from a singe viewpoint away from the frontal pose. Combined with visual motion tracking cues based on Kanade–Lucas–Tomasi (KLT) feature tracker [Lucas and Kanade,

1981] to enforce a temporal coherence and handle fast and natural head dynamics, we show that both accurate and robust head pose estimation can be achieved in natural and challenging scenarios as shown in Fig. 3.1c. In summary, the main contributions of this chapter includes:

- A 3D head representation combining the semantic and the precision of a 3DMM fitting and tracking under restricted poses with the robustness of a full head representation reconstructed from depth data. This includes the estimation and the maintenance of a fine pose correspondence between the 3DMM and 3D reconstruction.
- A symmetry regularizer for robust online 3DMM adaptation;
- A framework exploiting both visual motion tracking and 3D model semantics for frame-to-frame pose initialization;
- UbiPose, a dataset composed of 22 videos from the UBImpressed dataset [Muralidhar et al., 2016] featuring natural role played interactions, with more than 10k frames annotated with head pose ground-truth;
- Extensive experiments, with performance beyond the state-of-the-art on both the BIWI benchmark dataset and UbiPose.

3.2 Background on 3D Reconstruction

Reconstruction methods aiming at building 3D models of objects have attracted more attentions since the emergence of consumer 3D sensors. KinectFusion [Newcombe et al., 2011] is a standard approach which creates representations of static and rigid object or scene using a moving camera. Roughly speaking, it works by estimating the camera pose in each frame through the registration of successive scene observations, and projects the multi-angle viewed observations into a unified model representation for averaging. This approach has recently been extended via DynamicFusion [Newcombe et al., 2015] to also handle non-rigid objects through the estimation of a dense non-rigid warp field. Both KinectFusion and DynamicFusion rely on a volumetric representation which can be large and time consuming to process when aiming for precise reconstruction. To alleviate this problem, Keller et al. [2013] proposed a lighter point based reconstruction and fusion method, removing in the same way the static scene requirement through the robust detection of dynamic objects.

As we have seen, most previous model based methods are strongly focused on the face region. Although this is justified, as the main interest is on this region, it is nevertheless insufficient to address the large range of head pose variations observed in many natural human interactions situations of interest (cf. Fig. 3.6).

On the other hand, online reconstruction methods can potentially handle a large variety of poses, but are usually much more time consuming and have limitations. In particular they lack face and head semantic information and are more sensitive to fast motions. In addition,

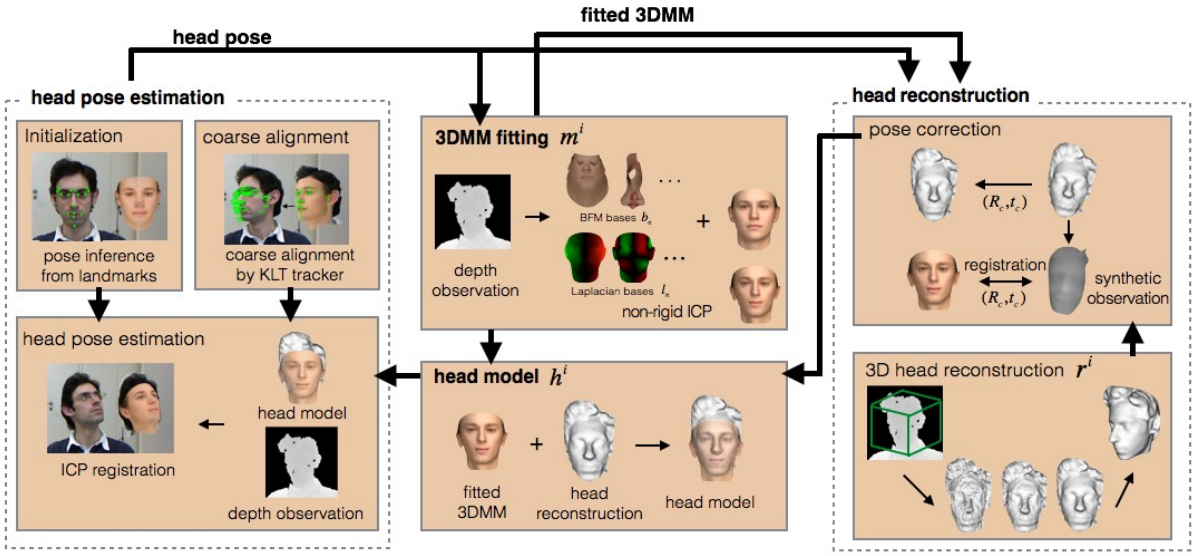


Figure 3.2: Proposed framework. The head pose estimation module registers the current head model **h** to the observations. The 3DMM fitting module personalizes a 3DMM face model **m** to sample frames online. The reconstruction module aggregates pose rectified depth images into a full head representation **r**. Vertex samples from the 3DMM face models **m** and from the reconstructed one **r** are used to define the head model **h**.

as faces and heads are not rigid, one could wonder how well such methods can work when being applied to natural interaction data with talking or facial expressions, or if the head shape is actually sufficient to obtain a precise registration when the face is almost not visible.

Therefore, this chapter propose a model combining the strengths of both approaches, through the online fitting of a 3DMM to the face region whereas the subject specific head representation is augmented on-the-fly through a variant of KinectFusion [Newcombe et al., 2011].

3.3 Method Overview

The proposed framework is illustrated in Fig. 3.2. It consists of three main modules: head pose estimation, 3DMM fitting and head reconstruction. The head pose estimation module aligns, at every time step i, the current estimate of the head model $\mathbf{h}^i$ with the observed RGBD data $(I^i, \mathbf{o}^i)$ (in which I denotes the RGB image and $\mathbf{o}$ denotes the depth map) using the Iterative Closest Point (ICP) [Besl and McKay, 1992; Chen and Medioni, 1992] algorithm. This module also exploits two other submodules for initialization: one relying on face detection and landmark localization to initialize tracking; a second one relying on visual KLT tracking for coarse pose temporal alignment from the previous frame, allowing to handle the fast head acceleration motions regularly observed in natural sequences.

The aim of the 3DMM fitting and head reconstruction modules is to learn and update the

head model $\mathbf{h}^i$ of the given person using the sequence of past observations. This is achieved using two main representations: the first one, $\mathbf{r}^i$, is a 3D reconstruction of the head obtained through the temporal registration and integration over time of the incoming depth frames. Its main advantage is that it can represent the full head without any prior knowledge. The second one is a 3D mesh face representation, $\mathbf{m}^i$, built and adapted online from a multi-instance 3DMM fitting algorithm relying on automatically selected depth frames.

Note that, in the following sections, we will refer to a representation "$\mathbf{h}$" as a set of vertices $\mathbf{v_h} := \{\mathbf{v_h}[k]\}_{k=1}^{N_v^\mathbf{h}}$ and normals $\mathbf{n_h} := \{\mathbf{n_h}[k]\}_{k=1}^{N_v^\mathbf{h}}$, such that $\mathbf{h} := \{\mathbf{v_h}, \mathbf{n_h}\}$. We will use this notation to refer to the different face representations $\mathbf{h}$, $\mathbf{m}$ and $\mathbf{r}$, while using the "$[k]$" to index specific vertices or normals. In this view, the resulting head model, used for pose estimation, is thus given by the joint set of vertices and normals coming from the two representations, i.e., $\mathbf{h}^i = \{\mathbf{m}^i, \mathbf{r}^i\}$.

While in principle after several frames we could rely only on the reconstructed model $\mathbf{r}$ for head pose estimation, we keep the 3DMM-based face model $\mathbf{m}$ as part of the head representation as it has several advantages. First, the semantic meaning of vertices from $\mathbf{m}$ is well known, which can be useful for face analysis, or if we want to further combine the model with appearance information provided by facial landmark detectors. Secondly, besides personalization of the face model to specific individuals, the 3DMM-based face model can be extended to include further elements, e.g. deformations due to expressions, which could be useful for further facial analysis. Thirdly, the existence of the 3DMM can prevent possible tracking failures caused by the sudden emergence of face regions which had not been seen so far and are thus not yet reconstructed, thus regularizing the resulting model.

Note that both the 3DMM-based face model and the head reconstructions are built online without any manual intervention. Details of pose estimation and head representation learning are provided in the following sections.

3.4 Model based 3D Head Pose Estimation

The objective of this module is to estimate the 3D head pose $\mathbf{p}^i = (\mathbf{R}^i, \mathbf{t}^i)$ at time i from the RGBD map $(I^i, \mathbf{o}^i)$. Here $\mathbf{p}^i$ represents a rigid transformation relating the head coordinate system (in which $\mathbf{h}$ is defined) to the world coordinate system, parameterized by a rotation matrix $\mathbf{R}^i \in R^{3\times3}$ (also characterized by three rotation angles yaw, pitch and roll), and a translation matrix $\mathbf{t}^i \in R^{3\times1}$.

The head pose estimation problem is formulated as finding the transform $\mathbf{p}^i$ of the head model $\mathbf{h}^i$ which minimizes the surface alignment error to the depth observations $\mathbf{o}^i$. However, this is intractable, as it requires to estimate jointly the pose and the point-wise semantic alignment between the surfaces. Thus, the cost is usually minimized using some form of ICP algorithm.

In the following, we first present in Section 3.4.1 the approach for ICP-based head pose

estimation. Since being trapped in local minima is a typical weakness of ICP, in Section 3.4.2 we describe our method to initialize ICP close to the target pose either in the first frame (tracking initialization) or from frame-to-frame (during tracking).

3.4.1 Pose Estimation.

Pose estimation is achieved using a variant of ICP, i.e., minimizing the registration error by iterating between the correspondence search and the rigid pose estimation steps.

More precisely, at each iteration, we first find the vertex correspondences of the head model, rigidly transformed by the current pose estimate, to the data $\mathbf{o}^i$ using the method in [Park and Subbarao, 2003], which is a fast implementation of normal shooting. We will denote $c^i(k)$ the index of the vertex in $\mathbf{o}^i$ found to be the correspondence of the vertex k in $\mathbf{h}$. Then the pose estimate is improved by minimizing the point-to-plane ICP cost $E_1(\mathbf{R}^i, \mathbf{t}^i)$ given by:

$$\sum_k w[k] \left((\mathbf{R}^i \mathbf{n}^i_{\mathbf{h}}[k])^{\mathrm{T}} (\mathbf{R}^i \mathbf{v}^i_{\mathbf{h}}[k] + \mathbf{t}^i - \mathbf{v}^i_{\mathbf{o}}[c^i(k)]) \right)^2 \tag{3.1}$$

where the time index i indicate that the set of vertices $\mathbf{v}^i_{\mathbf{h}}$ and normals $\mathbf{n}^i_{\mathbf{h}}$ refer to the head model $\mathbf{h}$ at time i.

The robust weights $\{w[k]\}_k$ aim to discard bad correspondences. Assuming $\delta[k]$ is the Euclidean distance between a transformed vertex and its correspondence, $w[k]$ is computed at each ICP iteration as follows: i) $w[k]$ is set to zero for correspondences whose normals differ by more than 45°, or if $\delta[k] > 4cm$; ii) $w[k]$ is 1 for $\delta[k] < 1cm$; iv) otherwise, $w[k] = \frac{r_1}{(\delta[k]-r_2)^2}$, where r_1 and r_2 are two parameters controlling the weight decay. We use the same weighting strategy for all ICP methods in this work.

3.4.2 Pose Initialization.

Initializing the ICP algorithm is needed in two distinct situations: to start the tracking of a newly detected head (or restart it upon a detected tracking failure); and during tracking, given the output result from the previous frame. Below we describe these two cases.

Tracking Initialization.

In most applications, the tracking may have to start with any pose from the head. To do so, we initialize the system by inferring the head pose from facial landmark detections. The toolkit Dlib [King, 2009] is used to detect the face and facial landmarks from the RGB image I^i using the method of [Kazemi and Sullivan, 2014]. These landmarks are then back-projected into the 3D space using the depth map $\mathbf{o}^i$ and used to form one-to-one point pairs with the corresponding 3D landmarks of $\mathbf{m}$, whose indices are known from the 3DMM semantics. Then the rigid rotation and translation of the head are inferred from these 3D point pairs using the

method in [Besl and McKay, 1992].

Temporal Coarse Alignment.

A common strategy in tracking is to use the pose estimated in frame $i-1$ as prediction and then initialization for frame i, or to use a more complex state-based dynamic model. These strategies have nevertheless difficulties in case of sudden acceleration (lagging behind) or deceleration (over shooting). A better strategy, as demonstrated in other tracking framework (e.g. GAVAM [Morency et al., 2008]) is to exploit visual motion for prediction. More precisely, a coarse alignment between frame $i-1$ and i is conducted based on facial feature tracking. Concretely, 3D facial features $\{\mathbf{f}^{i-1}[l]\}_l$ (where l denotes the feature index) of the head model set with the estimated pose of frame $i-1$ are projected into the 2D image plane I^{i-1}. Their corresponding positions in the next frame are estimated using a robust variant of the KLT tracker and projected back into the 3D space, resulting in the 3D feature locations $\{\tilde{\mathbf{f}}^{i}[l]\}_l$ predictions. The relative pose transformation $(\tilde{\mathbf{R}}^{i-1,i}, \tilde{\mathbf{t}}^{i-1,i})$ between frame $i-1$ and frame i is then estimated from the set of paired features $\{(\mathbf{f}^{i-1}[l], \tilde{\mathbf{f}}^{i}[l])\}_l$ by minimizing the following cost function:

$$\begin{aligned}&\sum_l w[l]\left(\left(\tilde{\mathbf{R}}^{i-1,i}\cdot\mathbf{n}^i_{\mathbf{f}}[l]\right)^{\mathrm{T}}\left(\tilde{\mathbf{R}}^{i-1,i}\mathbf{f}^{i-1}[l]+\tilde{\mathbf{t}}^{i-1,i}-\tilde{\mathbf{f}}^{i}[l]\right)\right)^2\\&+\gamma||\tilde{\mathbf{R}}^{i-1,i}-\mathbf{R}_I||^2,\end{aligned} \tag{3.2}$$

where $\mathbf{R}_I$ denotes the identity matrix and $\mathbf{n}^i_{\mathbf{f}}[l]$ is the normal vector at the l^{th} feature on the head model. The weight $w[l]$ is derived from the tracking confidence of the l^{th} feature given by the robust KLT tracker. A regularizer for the rotation matrix is incorporated to favor the identity rotation matrix estimation and comparatively encourage large translations in the solution, if required, since large and fast head motions are often due to head translation. Finally, given the relative pose transformation $(\tilde{\mathbf{R}}^{i-1,i}, \tilde{\mathbf{t}}^{i-1,i})$, a coarse estimation of the head pose at frame i is given by:

$$\mathbf{R}^i = \tilde{\mathbf{R}}^{i-1,i}\cdot\mathbf{R}^{i-1}, \mathbf{t}^i = \tilde{\mathbf{R}}^{i-1,i}\cdot\mathbf{t}^{i-1}+\tilde{\mathbf{t}}^{i-1,i} \tag{3.3}$$

In our implementation, we chose 70 facial landmarks and 80 random points on the head model for coarse alignment. We expect this to achieve a good balance between covering a wider face area (through random points) and points which normally result in high confidence motion estimates (facial landmarks).

3.4.3 Tracking Failure Identification.

In some situations the ICP optimization may diverge. If detected, we denote it as a tracking failure and apply the Dlib library face detector to incoming frames until a face is detected and the tracking is reinitialized. In this work, a tracking failure is identified using the weights $w[k]$. Concretely, we first select the visible points k_o from the aligned model. Then their weights are summed up as $\sum_{k_o} w[k_o]$ which indicates whether the registered model achieves an overall good correspondence with the observation. More precisely, if

$$\sum_{k_o} w[k_o] < 0.01 \cdot \sum_{k_o} 1 \tag{3.4}$$

is verified, then we assume that the registered model does not align with the observations, and a tracking failure is detected.

3.5 Person-Specific Face Modelling and Head Reconstruction

3.5.1 3D Morphable Model (3DMM) Fitting

In this section we explain our approach to retrieve $\mathbf{m}$ from a 3DMM. We will describe 3DMMs in detail, the fitting algorithm, the regularization, and the online sample selection strategy.

3D Morphable Model (3DMM).

A 3DMM is a statistical linear representation of facial shape (and/or appearance) variations [Vetter and Blanz, 1998]. Concretely, it is a linear combination of a mean shape μ with a deformation basis $\mathbf{b}_n$ weighted by a set of coefficients α. Vertex-wise, this can be represented as follows:

$$\mathbf{v}_\mathbf{m}(\alpha) = \mathbf{v}_\mu + \sum_{n=1}^{N_\mathbf{b}} \lambda_{\mathbf{b}_n} \alpha_n \mathbf{v}_{\mathbf{b}_n}, \tag{3.5}$$

where we omit the index notation "[k]" to avoid clutter and $\lambda_{\mathbf{b}_n}$ is the eigenvalue associated to the deformation vector $\mathbf{b}_n$. Here, $\mathbf{b}_n$ models facial shape variations across different face identities, while α allows to encode a person-specific facial shape $\mathbf{m}$.

In this work we used the Basel Face Model (BFM) [Paysan et al., 2009] as 3DMM. The deformation basis were learned from the 3D face scans of only 200 people, providing mainly global face variations. To obtain a finer modelling of a specific person's face, we rely on the work of [Bouaziz et al., 2013] which used the eigenvectors of the Laplacian matrix of the 3DMM graph as additional deformation bases. These Laplacian eigenvectors $\mathbf{l}_n$ corresponds to the smallest K eigenvalues, as shown in the 3DMM fitting module of Fig. 3.2 where the red and green region denotes positive deformation values and negative deformation values respectively, and the brightness of the region is proportional to the absolute deformation values. By adding the

Laplacian eigenvectors, our final 3D face model is given, per each vertex k, by:

$$\mathbf{v}_{\mathbf{m}}(\alpha,\beta) = \mathbf{v}_{\mu} + \sum_{n=1}^{N_{\mathbf{b}}} \lambda_{\mathbf{b}_n} \alpha_n \mathbf{v}_{\mathbf{b}_n} + \sum_{n=1}^{N_{\mathbf{l}}} \lambda_{\mathbf{l}_n} (\beta_n^x \mathbf{v}_{\mathbf{l}_n}, \beta_n^y \mathbf{v}_{\mathbf{l}_n}, \beta_n^z \mathbf{v}_{\mathbf{l}_n}) \tag{3.6}$$

Note that unlike $\mathbf{b}_n$, the Laplacian eigenvectors are the same across the directions x, y, z.

Online Model Fitting.

Since pose estimation is defined as a registration task aligning the head model to the observations, the head model itself should gradually be deformed to be as close as possible to the observations, and therefore adapt to the tracked person. To achieve this, we rely on a non-rigid multiple instance fitting method [Funes-Mora and Odobez, 2012] minimizing the discrepancy between our 3DMM model $\mathbf{m}(\boldsymbol{\alpha})$ and a set of frames $\mathscr{J}^i$ collected until time i. As with pose estimation, this discrepancy is minimized iteratively by minimizing the non-rigid ICP cost (with $(\mathbf{R},\mathbf{t}) = \{(\mathbf{R}^j,\mathbf{t}^j), j \in \mathscr{J}^i\}$):

$$E(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{R},\mathbf{t}) = \sum_{j\in\mathscr{J}^i} E_j(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{R}^j,\mathbf{t}^j) + \gamma_1 \sum_{n=1}^{N_{\mathbf{b}}} \alpha_n^2 + \gamma_2 \sum_{a\in\{x,y,z\}} \sum_{n=1}^{N_{\mathbf{l}}} \left(\beta_n^a\right)^2, \tag{3.7}$$

where the cost E_j for each sample j is given by:

$$E_j(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{R}^j,\mathbf{t}^j) = \sum_k w^j[k] \left(\left(\mathbf{R}^j \mathbf{n}_{\mathbf{m}}(\alpha,\beta)[k] \right)^{\mathrm{T}} \left(\mathbf{R}^j \mathbf{v}_{\mathbf{m}}(\alpha,\beta)[k] + \mathbf{t}^j - \mathbf{v}_{\mathbf{o}}^j[c^j(k)] \right) \right)^2, \tag{3.8}$$

meaning Eq. 3.7 represents the sum of the rigid alignment errors with each frame of the sample set $\mathscr{J}^i$, and a regularization over the coefficients (α, β). $\gamma = (\gamma_1, \gamma_2)$ are stiffness parameters controlling how much $\mathbf{m}$ can deviate from the mean shape. The solution of Eq. 3.7 is found using the Gauss-Newton method by gradually reducing the stiffness [Amberg et al., 2008].

Symmetry Regularizer.

The deformation basis, especially the Laplacian basis $\mathbf{l}_n$, do not generate a symmetric deformation fields over the face. When the 3DMM fitting is based on samples where some parts of the face are not observed, the fitting may be conducted locally on the visible parts and the resulting deformation fields may diverge on the not visible ones. We show some fitting samples in Fig. 3.3b which are based on the profile face samples in Fig. 3.3a. As can be seen, the resulting meshes are asymmetric and distort the original face shape, especially in the second case which is also affected by the long hair near the face. To address this issue, we designed a symmetry regularizer to penalize the deformation coefficients which provide asymmetric

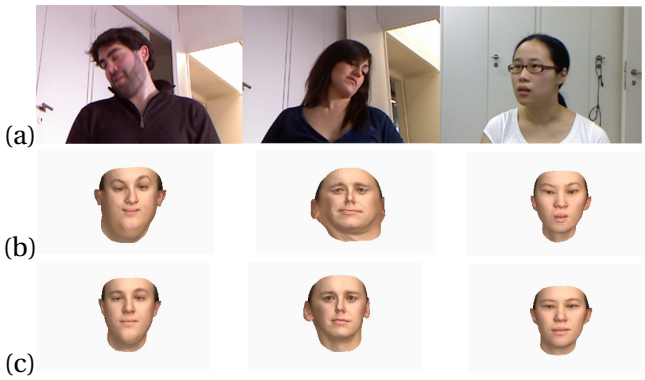


Figure 3.3: Use of symmetry regularizer.(a) Fitting samples; (b) Fitting without symmetry regularizer; (c) Fitting with symmetry regularizer

structure on the iteratively fitted face. This extends Eq. 3.7 as follows:

$$
\begin{aligned}
E(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{R},\mathbf{t}) = & \sum_{j\in\mathscr{J}^i} E_j(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{R}^j,\mathbf{t}^j) + \gamma_1 \sum_{n=1}^{N_\mathbf{b}} \alpha_n^2 + \gamma_2 \sum_{a\in\{x,y,z\}} \sum_{n=1}^{N_\mathbf{l}} \left(\beta_n^a\right)^2 \\
& + \gamma_3 \sum_k \left(\mathbf{v}_\mathbf{m}^x(\alpha,\beta)[k] + \mathbf{v}_\mathbf{m}^x(\alpha,\beta)[s(k)]\right)^2 \\
& + \gamma_3 \sum_k \left(\mathbf{v}_\mathbf{m}^y(\alpha,\beta)[k] - \mathbf{v}_\mathbf{m}^y(\alpha,\beta)[s(k)]\right)^2 \\
& + \gamma_3 \sum_k \left(\mathbf{v}_\mathbf{m}^z(\alpha,\beta)[k] - \mathbf{v}_\mathbf{m}^z(\alpha,\beta)[s(k)]\right)^2
\end{aligned}
\tag{3.9}
$$

where $s(k)$ denotes the symmetry index of vertex k. This mapping is derived from the original BFM model. For a symmetric BFM model, two symmetric points k and $s(k)$ should have the opposite x-axis values while the same y-axis and z-axis values. So the main idea of this regularizer is to maintain the symmetry of the face during progressive fitting, especially in absence of observations for some parts of the face. The fitting results using the symmetry regularizer are depicted in Fig. 3.3c, where better results are achieved than in Fig. 3.3b. Note that by preventing the fitting over the asymmetric long hair, the fitting in the second case is also improved.

Sample Set Online Selection.

A simple scheme is used to built $\mathscr{J}$ online. In essence, the goal is to collect observation samples whose estimated poses are close to 9 predefined poses [Funes-Mora and Odobez, 2012] (see Fig. 3.4), and to guarantee that the observation samples cover the whole 3D face. Whenever a new frame arrives, its pose is estimated using the current head/face model and the closest of the predefined poses is identified. If the estimation is in the neighborhood of the closet predefined pose and no frame had been yet added to that predefined pose, the current frame is added to form $\mathscr{J}^i$, and the model fitting optimizing Eq. 3.7 is conducted with all samples in $\mathscr{J}^i$. Note that as the number of samples increases, the value of γ further decreases to allow more flexibility for the fitting.

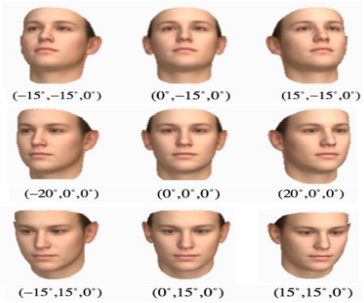


Figure 3.4: Set of predefined poses (yaw,pitch,roll) used to collect data samples for online 3DMM fitting.

3.5.2 Head Reconstruction Modelling

To handle head tracking from any pose, our goal is to dynamically augment the 3DMM-based mesh $\mathbf{m}$ with a head reconstruction built from the observed data. To achieve this, we rely on an adaptation of KinectFusion [Newcombe et al., 2011]. KinectFusion originally targets scenarios where a cameras moves in the 3D space or around a rigid 3D object. Our case is slightly different, as the sensor is static, and the head is moving.

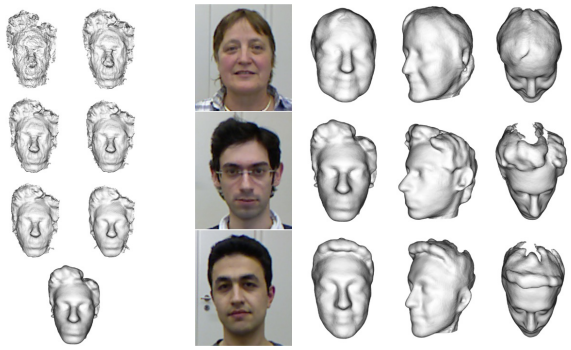
The principle is to represent the head through a 3D dense volume, composed of regularly samples voxels $\mathbf{v}_g$, and to accumulate observations using a truncated signed distance function TSDF[g], indicating which of the points g are inside (negative value) or outside (positive value) the head surface. We here use a 3D volume of 28($depth$) × 28($height$) × 19($width$)cm sampled with 128 steps per dimension.

The method comprises 4 main steps. The first one consists in estimating the head pose $(\mathbf{R}^i, \mathbf{t}^i)$. We rely on the robust method described in Section 3.4. Interestingly, this benefits from the availability of the 3DMM, esp. at the beginning when only few frames have been observed. The second step is the volumetric mapping, which consists in rotating the vertex samples in the camera pose according to $\mathbf{v}_g^i = \mathbf{R}^i\mathbf{v}_g + \mathbf{t}^i$. The third step consists of computing the per-frame TSDF [Curless and Levoy, 1996] associated to the observed surface, denoted as tsdf^i and defined by:

$$\text{tsdf}^i[g] = \frac{[\mathbf{v}_g^i]_z - [\pi(\mathbf{v}_g^i)]_z}{\tau} \tag{3.10}$$

in which $\pi(\mathbf{v}_g^i)$ denotes the projection along the ray of the vertex $\mathbf{v}_g^i$ onto the observed 3D surface, and $[\cdot]_z$ denotes the depth of a 3D point. In other words, tsdf records for vertex $\mathbf{v}_g$ the signed distance between its actual location $\mathbf{v}_g^i$ and the observed surface point. The parameter τ represents the thickness around the observed surface for which such distance is computed, and actually used (see equation 3.11 below).

Finally, in the fourth step, the tsdf values across frames are aggregated using a simple averaging



(a) online process (b) head reconstruction results

Figure 3.5: 3D head reconstruction from the BIWI dataset.

strategy:

$$\mathrm{w}_{\mathrm{ts}}^{i}[g] = \begin{cases} 1 & \text{if } -1 < \mathrm{tsdf}^{i}[g] < +\infty \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

$$\mathrm{TSDF}^{i}[g] = \frac{\mathrm{w}_{\mathrm{TS}}^{i-1}[g]\mathrm{TSDF}^{i-1}[g] + \mathrm{w}_{\mathrm{ts}}^{i}[g]\mathrm{tsdf}^{i}[g]}{\mathrm{w}_{\mathrm{TS}}^{i-1}[g] + \mathrm{w}_{\mathrm{ts}}^{i}[g]} \tag{3.12}$$

$$\mathrm{w}_{\mathrm{TS}}^{i}[g] = \mathrm{w}_{\mathrm{TS}}^{i-1}[g] + \mathrm{w}_{\mathrm{ts}}^{i}[g] \tag{3.13}$$

Importantly, note that the fusion is only conducted on voxels whose tsdf values are within the range $[-1,+\infty]$ (pixels in front of the surface or close behind the observed surface). This is to avoid self-occlusion effects for concave parts, e.g. when seen from 45°, the visible nose surface hides other face surfaces which might not necessarily lie 'inside' the head.

Reconstruction Model.

At each time step, a reconstruction model $\mathbf{r}^{i}$ is built as a 3D mesh from $\mathrm{w}_{\mathrm{TS}}^{i}$. More concretely, the marching cubes method [Lorensen and Cline, 1987] is applied to the set of voxels for which $\mathrm{w}_{\mathrm{TS}}^{i}$ is larger than 25 (i.e. voxels having been observed at least 25 times within the observed surface region) to find the zero crossing surfaces and extract the vertices and their normals. Examples of reconstruction results at the end of the sequence from the BIWI dataset are shown in Fig. 3.5, and demonstrate that accurate models can be recovered.

3.5.3 Head Model

As described in Section 3.4, what we need for pose estimation is a set of vertices and normals, i.e. $\{(\mathbf{v}_{\mathbf{h}}^{i}[k], \mathbf{n}_{\mathbf{h}}^{i}[k]), k = 1 \ldots N_{v}^{\mathbf{h}}\}$. To combine the 3DMM-fitted mesh $\mathbf{m}$ and the reconstruction model, we simply sample a fixed ratio of vertices from each of the model. That is, if $N_{v}^{\mathbf{m}}$ represents the number of vertices in $\mathbf{m}$, we randomly sample $N_{v}^{\mathbf{r}} = \eta \times N_{v}^{\mathbf{m}}$ from $\mathbf{r}$, and hence we have $N_{v}^{\mathbf{h}} = N_{v}^{\mathbf{r}} + N_{v}^{\mathbf{m}}$.

3.5.4 Pose Bias Correction

The 3D head reconstruction is a process which fuses the depth observations into a grid. To register the observations of different poses into a unified model, the grid is transformed with the estimated pose at every frame. Therefore, the estimated head pose is essential to the quality of reconstruction and needs to be consistent across frames.

However, in the initial frames, the pose estimation relies on the 3DMM-based representation **m**, which is progressively fitted to the person's face. If this fit is good (which is usually the case when starting the sequence from a near frontal pose), the estimated pose will be very close to the real one, and the reconstruction will then implicitly be built and aligned with **m**. However, if this fit is not yet fine, and/or if the initial estimated poses are biased, the reconstruction will be performed in a pose coordinate system slightly different than that of **m**, and this difference may remain over time. In other words, the two head representations (**m** and reconstruction) are not fully aligned, the same facial feature may appear on two different positions, and this can confuse the registration.

A pose correction module aligning the reconstruction with **m** is necessary. It mainly requires to estimate the pose bias between the two representations and then use this bias to align in the same pose space the vertices sampled from them when building the common head representation.

To estimate the bias we simply rely on ICP registration. However, to achieve a fast correspondence search [Park and Subbarao, 2003], we do not attempt at performing ICP directly between **m** and the 3D mesh of the reconstruction **r**. Instead, we simply render a synthetic depth image **s** by projecting the vertices $\mathbf{v_r}$ from the reconstruction into the depth image plane (i.e. along the ray of the depth camera). Thanks to the depth averaging during reconstruction, most temporal occlusions and large depth noise are removed and the resulting images are usually of high quality. Then ICP is performed to register the 3DMM model to this map and obtain the resulting pose correspondence bias $(\mathbf{R}^c, \mathbf{t}^c)$.

Finally, to account for this bias, two modifications need to be done. First, when building the head model (section 3.5), the vertices $(\mathbf{v}_\mathbf{r}^i[k], \mathbf{n}_\mathbf{r}^i[k])$ sampled from the reconstruction need to be mapped back into the semantic pose space of the 3DMM, according to

$$\begin{aligned} \mathbf{v}_\mathbf{r}^i[k] &\leftarrow \mathbf{R}^c \mathbf{v}_\mathbf{r}^i[k] + \mathbf{t}^c \\ \mathbf{n}_\mathbf{r}^i[k] &\leftarrow \mathbf{R}^c \mathbf{n}_\mathbf{r}^i[k] \end{aligned} \tag{3.14}$$

Secondly, in order to keep the consistency with the previous tsdf in the head reconstruction (Section 3.4), the volumetric mapping needs to be the composition of the estimated pose $(\mathbf{R}^i, \mathbf{t}^i)$ and the inverse correction $(\mathbf{R}^c, \mathbf{t}^c)$.

Importantly, note that since the reconstruction and 3DMM-fitted model evolve slowly after the initial frames, the pose correction $(\mathbf{R}^c, \mathbf{t}^c)$ is only computed every 100 frame in our

implementation (but it is used at all frames).

3.6 UbiPose Dataset and Experimental Protocol

In this section, we present the design of our experiments, including the datasets and ground truth, the performance measures, the considered models along with parameter settings.

3.6.1 Dataset

In our experiments, two datasets are used.

The BIWI Dataset

It is a public dataset collected by Fanelli et al. [2013]. It consists of 24 videos (15K frames in total) recorded with a Kinect 1 sensor, and where seated people keep moving their heads in an artificial fashion. Some samples are shown in Fig. 3.6a.

The UbiPose Dataset.

This dataset relies on videos from the UBImpressed dataset, which has been captured to study the performance of students from the hospitality industry at their workplace [Muralidhar et al., 2016]. The role play happens at a reception desk, where a student has to handle a difficult client. Students and clients are recorded using a Kinect 2 sensor (one per person). In this free and natural setting, large head poses and sudden head motions are frequent as people are observed from a relatively large distance, and people are mainly seen from the side (see Fig. 3.6b for samples).

Out of the 160 interactions recorded in the UBImpressed dataset, we randomly selected 22 videos (with 22 different persons) as evaluation data to build the UbiPose dataset. In 10 of these videos, 30-50 second clips were cut from the original videos and all frames were annotated (see Section 4.2.2). The other 12 videos were fully annotated at one frame per second. This allowed to gather a large diversity of situations. In total, this amounts to 14.4K frames. The UbiPose dataset with annotations and evaluation code can be found at www.idiap.ch/dataset/ubipose.

3.6.2 Ground Truth

BIWI Data

This dataset provides the ground truth of head pose $(\mathbf{R}, \mathbf{t})$ for every single frame, which was estimated using a supervised 3D face fitting and registration process. Fig. 3.7a indicates the distribution of the number of frames over pose ranges for this dataset

Figure 3.6: Dataset samples. a) BIWI. b) UbiPose.

UbiPose Data: Inferred Pose Ground Truth (IGT)

To avoid interfering with the role play, no wearable sensors, e.g. motion capture, were used to obtain a head pose ground truth. So we inferred the ground truth indirectly from facial landmarks. Concretely, we first annotated 6 landmarks on the extracted RGB frames whenever they were visible: left and right corner of the left eye (l-l and r-l), left and right corner of the right eye (l-r and r-r), nose tip (n-t) and nasal root(n-r). Generally speaking, these landmarks are rigid and seldom affected by facial expressions.

These 2D landmark annotations were projected into the 3D space using the depth image, and further paired with the corresponding landmarks in the 3DMM. Note that to foster precise head pose ground truth, the 3DMM had been previously fitted to the person's face using auxiliary data from a recording session made another day (with an interview scenario). The ground truth was then inferred from the available point pairs [Besl and McKay, 1992]. We denote the inferred ground truth as IGT.

Since for near profile poses the IGT might become noisy (see Section 4.4 for an evaluation of the annotation accuracy), we resorted to visual inspection to validate the IGT. More precisely, we examined the IGT frame by frame by projecting the IGT posed 3DMM model on the 2D image and compared it with the actual pose of the person in the image. If they were perceived as not matching, a new annotation was attempted. If the difference remained unacceptable after revision, the frame was definitely abandoned. After inspection, we obtained a dataset of 10.5K frames in total for experiment. The distribution of frames with respect to the head pose is illustrated in Fig. 3.7b. As can be seen, due to the scenario, most frames fall within a $[20°, 40°]$ interval. Compared with the BIWI dataset, we observe that there are much less frontal faces, whereas the percentage of frames above 80°is higher.

3.6.3 Performance Measurement

Head pose estimation performance can be evaluated by two aspects, *accuracy and robustness*. Below we describe how we measure them on the two datasets.

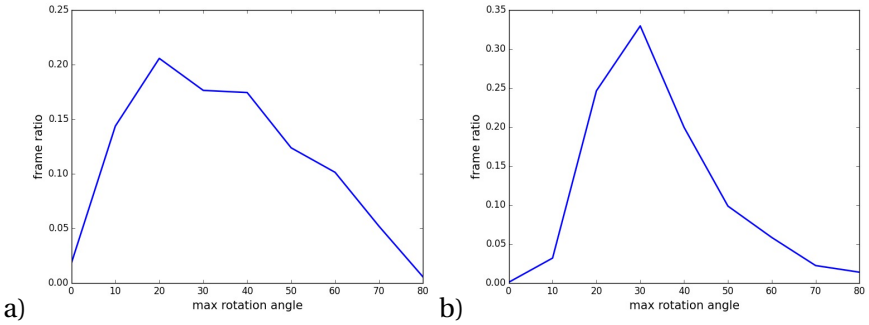


Figure 3.7: Proportion of frames with a given pose GT (or IGT for UbiPose) for (a) the BIWI dataset (b) UbiPose dataset.

Accuracy

Since we have head pose ground truth, we firstly report accuracy using on one hand the average error of the estimated rotation angles. On the other hand, since we have annotated up to six landmarks per frame on the UbiPose dataset, we also measure the accuracy of landmark localization. Note that the landmark location estimates are obtained by projecting the semantic 3DMM-fitted model set with the estimated pose, and the accuracy of landmarks is measured as the 2D distance between the estimated and annotated landmark locations.

Robustness

Robustness can be defined by several aspects. One of them is to evaluate whether the error can be kept in an accepted range even when extreme head pose occurs. This can be measured with the cumulative distribution function of errors (error CDF) showing the proportion of frames whose errors are below a given value. We further use this curve to report as in [Meyer et al., 2015] accuracy measures $\mathbf{ACC}_{10}$ as the percentage of frames with L2 norm of angular errors below 10 degrees.

We can also analyse the robustness by measuring the accuracy across different head poses. We take the maximum of the three ground truth rotation angles (yaw/pitch/roll) as pose indicator per frame and quantize them in bins of size 10°. Then the average error is computed within each bin.

The robustness also usually means continuous and successful tracking. Therefore, we define the lost frame ratio (**LR**) to indicate the percentage of frames in which the tracker is in a failure state, because a tracking failure has occurred and that the tracker could not successfully reinitialize itself. In addition, to better analyze the robustness to occlusion, we annotated video segments where at least half of the face is occluded and computed the ratio **O-LR** of tracking failure which have occurred in such segments. Note that **O-LR** is an event based measurement which complements the frame-based measure **LR**.

Finally, we also measure the impact of facial deformations on pose estimation. As a proxy for

Table 3.1: Average error of IGT.

	yaw	pitch	roll	mean±std	$\mathbf{ACC}_{10}$
IGT	3.26	3.79	2.48	3.18±1.61	100.0%

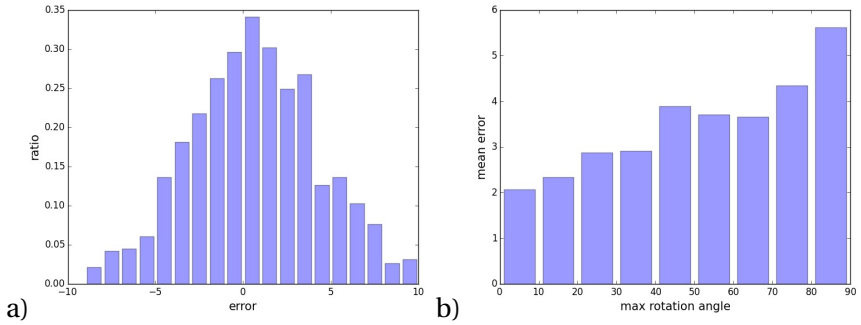


Figure 3.8: IGT quality evaluation based on BIWI data. a) distribution of the yaw, pitch and roll errors between IGT and GT. b) distribution of IGT absolute pose errors.

this, with the help of microphone array data (recorded as part of the UbiPose database), we extracted frames where the participants are speaking since facial deformations are expected to be important in these frames due to mouth motion. The results on these frames are reported in the column "**S-mean**" (mean on speaking frames) of the result tables.

Significance Test

To evaluate whether the results of two methods are statistically significant (esp. with respect to the proposed model), and given the large number of samples available, we rely on a paired z-test and report the significance of the test for different p-values.

3.6.4 IGT Evaluation

Compared with the supervised dense ICP registration of BIWI, the inferred ground-truth (IGT) is only based on limited facial landmarks. Even if we used a visual inspection to validate them, we have no clear idea about its accuracy. To evaluate this, we conducted a small scale experiment in which we annotated a subset of the BIWI dataset with landmarks, and following the same procedure than with the UbiPose data, inferred the IGT. In practice, we used a subset of 450 random frames, and 381 remained after visual validation.

Table 3.1 and Fig. 3.8 provide the results of the IGT evaluation. As can be observed, the mean error is around 3°, with a small standard deviation. Fig. 3.8a shows that errors follow a zero mean Gaussian distribution, indicating that no bias is observed. In addition, as was to be expected (less available landmarks, higher inference sensitivity to location accuracy), Fig. 3.8b shows the limited increase of error in function of the observed pose, but even for large pose, the average error remains below 6°. In general, around 85% of the errors are below 5°, and all

errors are below 10°.

Altogether, although not perfect, we believe that the IGT can be used as GT for UbiPose. While the reported error may not reflect the actual accuracy of methods, given the large number of samples (above 10000), which are dominantly independent and uncorrelated and with unbiased approximation, we expect the evaluation to provide a fair indication of which method performs best. This is particularly true for performance measures like $\mathbf{ACC}_{10}$ which are indicative of robustness, and somehow already account for some uncertainty in the ground truth. In any case, although of a different nature, the raw annotated landmarks will also be used for evaluation.

3.6.5 Systems and Parameter Settings

We compared several models as listed below:

- **Mean shape:** the tracking is conducted with ICP using the mean shape of the Basel Face Model (BFM).
- **3DMM:** An online model fitting is conducted using the BFM and Laplacian basis function and the symmetric constraint. This differs from most previous works which only fit the BFM model [Funes-Mora and Odobez, 2012; Meyer et al., 2015; Yu et al., 2017]. Also, following [Bouaziz et al., 2013], the 3DMM and Mean shape models rely on a sample subset of the vertices of the full BFM model, with a denser sampling on rigid face regions (forehead, eye regions) [Funes-Mora and Odobez, 2012; Bouaziz et al., 2013].
- **FWH-ID and FWH-EXP:** FaceWarehouse [Cao et al., 2014] is a 3D facial expression database providing aligned 3D head models of 47 expressions from 150 participants. We derived the deformation bases of both identity and expressions from these 3D scans and built two models, namely FWH-ID and FWH-EXP. For the FWH-ID model, only the identity deformation bases are used and the online model fitting is conducted as with the BFM model. This allows to evaluate the impact of the used 3DMM mesh model (BFM vs FWH) on the tracking results. In the FWH-EXP model, the expression bases are added to the identity bases of the FWH-ID model, allowing to test the performance of using a richer (and in principle more relevant) model to fit the data.
- **FHM:** the tracking is only based on the reconstructed head model, except in the first 25 frames where the 3DMM is still used to build an initial 3D model.
- **HeadFusion:** this is the proposed model. It includes both the online model fitting with Laplacian basis function and symmetric constraint, the KLT tracking initialization, head reconstruction with pose correction. The default value for η (proportions $\eta = \frac{N_v^{\mathbf{r}}}{N_v^{\mathbf{m}}}$ of points coming from the reconstructed $\mathbf{r}$ and 3DMM models, see Sec. 3.5.3) is set to 1.5.
- **State-of-the-art:** we compared our work with three methods. The 3DMM fitting based approach [Meyer et al., 2015] using Particle Swarm Optimization (PSO) for tracking,

which achieves the best results on BIWI; the OpenFace system [Baltrušaitis et al., 2016] which relies on both image and depth data and has been primarily optimized for landmark localization; and our previous work [Yu et al., 2017], which combines 3DMM and reconstruction but without several key elements like KLT, pose correction, symmetric fitting constraint.

Parameter Settings

All model parameters were kept the same for all experiments (except for reporting explicit changes, eg. the impact of η) and the two datasets. Whenever relevant, we used $N_{\mathbf{b}} = 50$ deformation bases from the BFM model and the same number for the Laplacian model. For all models involving head reconstruction, the size of the 3D volume is $128 \times 128 \times 128$.

3.7 Results

To analyse our results, we first present qualitative results in Section 3.7.1. We then detail numerically our results in Section 3.7.2, comparing the different head representation approaches, including against state-of-the-art methods. Finally, in Section 3.7.3, we evaluate the benefits of the different components of our method.

3.7.1 Qualitative Results

Fig. 3.9 illustrates the obtained results. As can be seen, robust and accurate tracking can be achieved, in both typical and more adverse conditions, like leaning, looking towards the back while calling on the phone, or putting the hand in front of the mouth. The impact and requirement for a full head representation is clearly visible, and our method allows to handle it, even when the head (eg the right three pictures in Fig. 3.9c) is only partially visible and could easily lead to uncertainty in pose estimation and tracking failure. In addition to the head representation, KLT tracking proved to be particularly useful, e.g. in handling people looking for objects in the registration desks, where non-frontal faces with fast motion and pose changes could be observed (Fig. 3.9c).

3.7.2 Quantitative Analysis

The tracking and head pose results of all methods are listed in Table 3.2 (BIWI) and Table 3.3 (UbiPose), whereas Table 3.4 display the results for the landmark localization task on UbiPose data. For further analysis of the models' properties, error CDF, error distribution on poses and **LR** distribution are also provided in Fig. 3.10. In the sequel, we will first analyze the performance of the different head pose modelling methods before comparing to the state-of-the-art.

Table 3.2: BIWI: average head pose error and accuracy. $^{\dagger}p<0.01$[1]

Approach	yaw	pitch	roll	mean (std)	$\mathbf{ACC}_{10}$
FWH-ID	2.47	3.01	2.15	2.54 (3.5)†	94.7%
FWH-EXP	2.46	3.87	2.15	2.83 (3.6)†	92.4%
Mean shape	5.20	2.72	4.23	4.05 (9.2)†	88.7%
3DMM	2.96	1.58	2.65	2.40 (4.8)†	94.7%
FHM	3.30	1.82	2.45	2.52 (3.2)†	94.5%
HeadFusion	2.54	1.45	2.10	2.03 (3.0)	96.4%
OpenFace [Baltrušaitis et al., 2016]	7.77	7.99	4.61	6.79 (6.8)†	52.3%
Yu et al. [2017]	2.49	1.53	2.18	2.07 (5.2)	96.6%
PSO [Meyer et al., 2015]	2.1	2.1	2.4	2.2	94.6%

Overall Result

We first compare the FaceWarehouse models with the BFM model. We first note that the FWH-ID model performs worse than the (BFM) 3DMM model. This might be explained by the fact that the BFM model was built from high resolution scans, compared to lower quality data for the FWH model (for which vertex subsampling was not necessary because of the lower resolution). Furthermore, we find that the performances (both for pose and landmark estimation) of the FWH-EXP expression model are worse than those of its ID only counterpart FWH-ID. This is not so surprising, as in presence of noise or non frontal head pose, the additional fitting capacity may lead to expression basis fitting pose or identity information rather than only the facial deformation, resulting in a distorted face model whose fitting reduces the accuracy of head pose estimation. In practice, as noted in [Bouaziz et al., 2013], to handle facial deformation, it is more efficient (and better) to first fit the head pose, and then estimate the facial deformation. In contrast, the expression independent BFM model achieves better performance in head pose estimation.

We then compare the BFM Mean Shape, 3DMM, FHM and our HeadFusion models. As can be seen, our proposed model HeadFusion has the best accuracy and robustness for most performance measures: it has the lowest head pose error on both BIWI and UbiPose, the lowest landmark localization error on UbiPose, and the best robustness indicators (least error variance, lowest **S-mean**, **LR** and **O-LR**), indicating that it has a more stable tracking and suffers from much less tracking failures. In particular, the tracking failure in occlusion cases of our method is much less than the approaches without reconstruction. This is understandable since our model can rely on more points from the full head for model registration. The robustness is also reflected in Fig. 3.10d where for large poses, **LR** is much lower for our approach.

Notice that the accuracy gap between the proposed models and the others is larger on UbiPose than on BIWI, probably because the former dataset involves more natural behaviours and comprises much more diverse and adverse situations. Note as well that for UbiPose dataset average errors and curves in Fig. 3.10 are reported on frames without failures, thus results from our approach are computed from more frames. This explains why the $\mathbf{ACC}_{10}$ on UbiPose

Figure 3.9: 3D head reconstruction and tracking samples. a) BIWI dataset. b) Typical frames of UbiPose dataset. c) Extreme head pose cases and occlusion cases. Note that for better visualization, displayed image were cropped from original images.

is better for the 3DMM than for our approach, since the 3DMM errors are gathered from less frames and in particular exclude those which often correspond to difficult situations and higher pose errors in general.

Looking at the **S-mean** results (speech frames with facial deformations) in Table 3.3 and in Fig. 3.10h, we can notice that almost all methods keep a stable performance compared with the overall results (mean), including under difficult poses. For the BFM based methods, this can be attributed to two main factors: first, our robust weighting strategy of ICP (see Sec. 3.4.1) which can filter out bad correspondences caused by facial deformations; secondly, the selection of

[1] † indicates that the result is significantly lower than our method with $p < 0.01$. The test with PSO [Meyer et al., 2015] is not possible.

Table 3.3: UbiPose: average head pose error and accuracy. †p<0.01

Approach	yaw	pitch	roll	mean (std)	**S-mean** (std)	**ACC$_{10}$**	**LR**	**O-LR**
FWH-ID	8.45	4.87	4.94	6.09 (9.7)†	6.22 (10.7)†	70.3%	4.1%	30.8%
FWH-EXP	11.55	6.65	7.16	8.45 (15.1)†	8.64 (15.5)†	64.7%	5.7%	46.2%
Mean shape	6.77	5.03	5.12	5.64 (7.5)†	5.75 (8.0)†	64.2%	3.5%	38.5%
3DMM	5.63	5.05	4.57	5.08 (6.9)†	5.10 (6.9)†	70.9%	4.1%	38.5%
FHM	5.33	4.96	4.61	4.97 (2.8)†	5.06 (3.3)†	56.0%	3.6%	15.4%
HeadFusion	4.63	4.37	3.83	4.28 (2.7)	4.25 (3.0)	70.0%	0.6%	15.4%
OpenFace [Baltrušaitis et al., 2016]	9.49	4.45	4.89	6.27 (4.0)†	6.94 (4.7)†	44.3%	8.7%	100.0%
Yu et al. [2017]	5.09	5.14	3.90	4.71 (4.5)†	4.60 (4.8)†	67.2%	3.3%	23.1%

Table 3.4: UbiPose: Landmark position errors. †p<0.01

Approach	l-l	r-l	l-r	r-r	n-r	n-t	mean
FWH-ID	9.6	9.1	10.2	9.5	13.3	13.9	11.3 (19.7)†
FWH-EXP	11.5	11.7	14.0	15.1	16.9	18.2	14.7 (27.8)†
Mean shape	7.7	10.8	10.7	11.1	9.3	10.1	9.7 (15.8)†
3DMM	7.5	10.9	10.1	11.5	8.9	10.5	9.6 (16.5)†
HeadFusion	5.4	7.9	7.1	9.3	6.0	6.5	6.7 (6.0)
OpenFace [Baltrušaitis et al., 2016]	5.1	5.8	5.6	6.6	6.0	6.0	5.8 (4.1)
Yu et al. [2017]	6.0	9.3	9.2	11.4	7.4	8.6	8.1 (12.1)†

mesh samples in face regions less affected by facial deformations. On their side, by averaging faces over time, reconstruction models (FHM or HeadFusion) result in a neutral model which combined with the previous factors, avoids the addition of specific facial expression biases.

All in all, these results demonstrate that our method has the potential for continuous and uninterruptive tracking which is necessary for tracking in natural interaction setting. This is due to the good exploitation of the joint benefit of the 3DMM model and of the FHM approach.

Indeed, on one hand, compared to FHM, the 3DMM achieves higher accuracy, as demonstrated by a much higher $\mathbf{ACC}_{10}$ of 70.9% compared to 56.0% on UbiPose, but this is at the cost of less robustness: a much higher variance and difficulty to detect tracking failure (thus reporting larger errors, as can be noticed from the fact that the CDF of the 3DMM does not reach 100% in Fig. 3.10e), in particular for large head pose (see Fig. 3.10b for instance).

On the other hand, the FHM model is less accurate (see the worse CDF curves at small angles on BIWI and more importantly on UbiPose, Fig. 3.10e), but is more robust as shown by the much smaller pose error standard deviation on UbiPose, or the lower tracking failure **LR** and **O-LR** compared to the 3DMM. However, it is clear from the results that the FHM model alone is not sufficient to achieve good tracking, and that it is the combination of the 3DMM and FHM which performs best.

Finally, regarding the Mean Shape model, one can notice that its results on BIWI and UbiPose (Tab. 3.2 and Tab. 3.3) are lower than other models including the 3DMM model. In fact, the error of the Mean shape model is relatively large for almost every pose bin according to Fig. 3.10b and f[2], which reflects the importance of online model adaptation in model registration.

[2]Note that in Fig. 3.10f, the error of the Mean shape in the first bin (<5°) is abnormally high. This is due to the fact that there are only 8 frames in that bin (see Fig. 3.6b), and for that method, the tracking results are bad for 6 contiguous frames due to the impact of an erroneous tracking right before these frames.

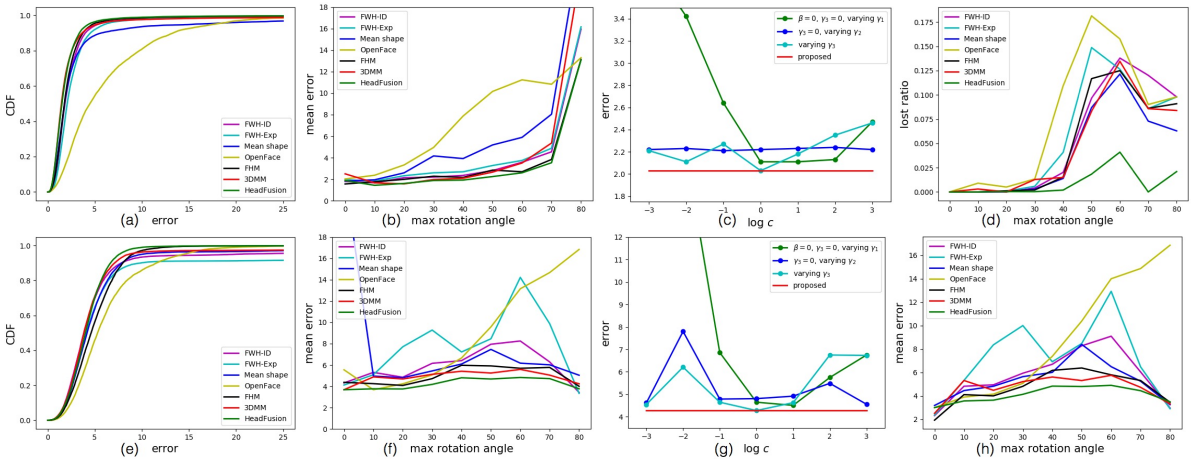


Figure 3.10: Robustness curves for BIWI (a,b,c) and UbiPose data (d,e,f,g,h). (a,e) Pose error CDF. (b,f) Mean error per pose. (c,g) Impact of regularization coefficients. (d) Tracking lost ratio per pose (UbiPose). (h) Mean error on speaking frames, per pose (UbiPose).

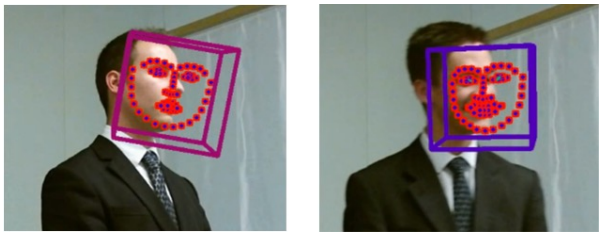


Figure 3.11: CMU OpenFace common failures. Although the error distance with respect to the visible landmarks is small, the head pose is very badly estimated. Note that OpenFace is using depth information as well.

Comparison with State-of-The-Art Methods

Three state-of-the-art methods were used, as described in Section 3.6.5: PSO [Meyer et al., 2015], the CMU OpenFace [Baltrušaitis et al., 2016], and our previous work [Yu et al., 2017].

BIWI Dataset. Our HeadFusion model obtains the best results. It exceeds the performance of PSO [Meyer et al., 2015] which relies on the combination of ICP and Particle Swarm Optimization (PSO), which shows that when combining a 3DMM with head reconstruction, ICP alone can achieve equal or even better accuracy. In particular, the estimation of the pitch angle is much improved compared to [Meyer et al., 2015]. OpenFace provides by far the worst error, which is understandable since it does not attempt at building a 3D face model. This shows the limitations of such approach for head pose estimation. Finally, on BIWI, we do not notice much improvement from our method compared to our previous work [Yu et al., 2017].

UbiPose Dataset. Table 3.3 demonstrates that our method performs much better than OpenFace for pose estimation, both in terms of accuracy and importantly robustness (much better $\mathbf{ACC}_{10}$, **LR** and **O-LR** values). This claim is further supported by Fig. 3.10f, which shows that the error of OpenFace becomes much larger beyond 45°. Note that the **O-LR** value is 100%,

Table 3.5: BIWI contrastive experiments.

Approach	yaw	pitch	roll	mean (std)	**ACC**$_{10}$
HeadFusion	2.54	1.45	2.10	2.03 (3.0)	96.4%
$\eta = 0.5$	2.48	1.41	2.19	2.03 (3.1)	96.4%
$\eta = 1.0$	2.56	1.46	2.15	2.06 (3.3)	96.5%
Without Correction	3.24	1.66	2.35	2.42 (5.8)	95.7%
Without KLT	2.98	1.78	2.34	2.37 (6.0)	96.0%
Without Sym	2.67	1.68	2.31	2.22 (4.4)	96.0%
Without Fitting	2.93	1.97	2.36	2.42 (3.6)	94.6%
Without KLT, Lap, Sym	2.75	1.74	2.37	2.29 (4.7)	95.8%

Table 3.6: UbiPose contrastive experiments.

Approach	yaw	pitch	roll	mean (std)	**ACC**$_{10}$	**LR**	**O-LR**
HeadFusion	4.63	4.37	3.83	4.28 (2.7)	70.0%	0.6%	15.4%
$\eta = 0.5$	5.51	4.62	4.56	4.90 (4.9)	69.5%	0.6%	7.7%
$\eta = 1.0$	4.88	5.02	4.12	4.68 (5.2)	69.5%	0.6%	15.4%
Without Correction	7.57	4.96	4.51	5.68 (10.5)	70.4%	0.7%	23.1%
Without KLT	4.72	4.37	3.88	4.33 (3.2)	70.0%	3.3%	23.1%
Without Sym	4.75	5.38	4.31	4.81 (5.1)	65.7%	0.7%	23.1%
Without Fitting	9.48	5.21	5.74	6.81 (9.8)	61.2%	0.6%	15.4%
Without KLT, Lap, Sym	9.39	5.99	5.49	6.96 (8.8)	58.3%	3.2%	23.1%

which shows that the OpenFace has difficulty in handling cases where at least half of the face is occluded. However, its performance for landmark localization is usually better, as shown in Table 3.4, as it was specifically trained for that[3]. This is not contradictory: since localization accuracy is computed only for visible landmarks, the localization errors can still remain small even if the pose estimate is bad, esp. for adverse situations where only a limited set of landmarks is visible. This contrast is illustrated in Fig. 3.11 and such situations are relatively frequent for OpenFace.

Compared to our previous work, the difference of the mean error is not that large (but is still statistically significant). However, the robustness is much higher with our new method, as shown by the higher **LR** and **O-LR** values of [Yu et al., 2017], which, without the coarse temporal alignment module, can not handle most fast motions of our data.

3.7.3 Model Components Analysis

In this section, we study the contribution of the different modelling components to the success of the method. We present and contrast the results of 7 experiments in Table 3.5 (BIWI) and Table 3.6 (UbiPose) by changing system parameters or removing some components.

Our approach samples points from the 3DMM and the 3D reconstruction to build the head model, with a ratio η. The default value in HeadFusion is $\eta = 1.5$, meaning that more points are sampled from the reconstruction. As can be seen, when using smaller values, the model is

[3]Remember however that due to the difference in tracking failures (**LR**) the average error of OpenFace is computed on 8.1% less frames than our method, frames in which the pose is usually large.

slightly less accurate, and less robust (higher error and standard deviation), in particular for UbiPose dataset.

Head Pose Correction.

Results show the requirement for this correction module. Without it, the error becomes larger and more variable, especially for the UbiPose dataset. This can be explained by the fact that sequences start with a semi-profile face, which usually result in a small but non negligible initial bias between the 3DMM model and the head reconstruction. Interestingly, the removal of the component does not seem to result in much more additional tracking failures.

Temporal Alignment.

When removing the coarse temporal alignment relying on the KLT tracker, we can notice that the performance does not decline too much in accuracy measurement. Rather, as shown by the results in Tab. 3.6, there is a higher number of tracking failures due to dynamical head motions that the tracker can not handle anymore.

Head Model Fitting.

The end of Section 3.7.2 highlighted the complementarity and mutual benefit of using the 3DMM and FHM head models. Here we further study the impact of the 3DMM modelling on results by removing some deformation bases in Eq. 3.9 ($\beta = 0$), or by varying the weights of the regularizing terms (γ_1, γ_2, γ_3) by a factor c ($c = 0.001, 0.01, 0.1, 10, 100, 1000$) with respect to their default value. Results are reported in Tab. 3.5 and 3.6, and Fig. 3.10c,g. We first remove both the symmetry regularizer and the Laplacian bases (green curve) and vary the weight γ_1. We observe that enforcing more ID shape bases regularization usually lead to better results. However, when γ_1 becomes too large, results quickly degrades in both datasets, and in practice, the fitted head models remain very close to the mean shape model. This is corroborated by results in Tab. 3.5 and 3.6 ('without Fitting'), which show that simply using only the 3DMM mean shape actually achieves worse results than methods with online model adaptation. When adding Laplacian bases and adjusting the weight γ_2 ($\gamma_3 = 0$) , we note that the performances are relatively stable for different values of γ_2. This is understandable, since the Laplacian bases mainly compensate the original deformation bases for a finer 3DMM modelling. However, the performances with Laplacian bases are inferior to the model with a suitable value of γ_1. Indeed, with fitting samples seen from semi-profile, a poor 3DMM fitting can be obtained (as already illustrated in Fig. 3.3) with asymmetric variations coming from Laplacian bases, for which a symmetry constraint is a must. Finally, we observe an improvement of performance (0.18 degrees on BIWI, 0.53 degrees on UbiPose) when using a suitable symmetric regularization. We also note from Fig. 3.10c,g that emphasizing too much on symmetry regularizer can make the 3DMM fitting too constrained and lead to worse

performances. Altogether, results in Fig. 3.10c,g show that our model with selected weights achieves the best compromise between robustness, accuracy, and quality of face fitting.

Finally, when removing the regularization and the KLT tracking, the negative effects are cumulated, resulting in a performance decrease in both accuracy and robustness.

3.7.4 Computational Cost

We implement our system in Python/C++ based on CPU. Generally speaking, the coarse temporal alignment based on KLT tracker takes ~60ms and the following ICP based alignment costs ~9ms. The 3DMM fitting executed in a separate thread usually costs ~5s. The reconstruction module which also includes the 3D meshing takes ~0.25s per frame. This module is applied at every frame within the first 300 frames and every 5 frames afterwards. The whole system can be much faster by implementing some modules (especially reconstruction) on GPU.

3.8 Conclusions and Future Works

We presented an accurate and robust 3D head pose estimation method effective even for challenging natural settings in this chapter. The main idea is to build a full head model providing more support when dealing with arbitrary tracking situations. To achieve this, we simultaneously conduct a 3DMM online fitting and and online 3D head reconstruction using a KinectFusion methodology. In addition, we also proposed a coarse temporal alignment module to handle fast head motions and a symmetry regularizer for finer model adaptation. Results demonstrate that our method achieves state-of-the-art performance and is also accurate and very robust when dealing with challenging natural interaction sequences where adverse situations are frequent.

Recovering the semantic segmentation of the head model (eg which region is face or hair) is an interesting perspective to the work. Indeed, the semantic information could help the landmark localization estimation to be more accurate; and secondly, as our method is still challenged by long hairs moving around, the knowledge of the semantic information could help in obtaining even more robust results. Our work can also be expanded to other tasks, by serving as a preprocessing step for head gesture recognition, eye gaze tracking, or facial expression estimation and analysis as shown by our experiments on this topic.

4 Head Nod Detection

We introduce our nod detection approach in this chapter. This approach is based on a full 3D face centered rotation model, and we propose to extract head rotation dynamics in head coordinate system and attempt to achieve body motion invariant nod detection by designing a feature related to the head rotation axis.

The rest of this chapter is organized as follows: we motivate our main idea and list the contributions in Chapter. 4.1; A brief method overview is summarized in Chapter. 4.2; The relative head transformation and feature extraction are covered in Chapter. 4.3 and Chapter. 4.4 respectively; Chapter. 4.5 and 4.6 demonstrates the experiment protocol and results; Finally, we make a summary of our approach in Chapter. 4.7.

4.1 Motivation and Contributions

As mentioned in Chapter. 2.2, although a number of works on head nod or head gesture detection have been proposed, many works are constrained to a specified head pose and it is hard to apply them to a scenario where the head poses are different than the ones used in the training data.

In our work, unlike previous approaches with 3D head trackers that extract angular velocities directly by differentiating the Euler angles obtained from the pose expressed in the camera coordinate system, we propose to calculate the relative rotation at each instant with respect to the head pose at some instance before. The Euler angles from this relative rotation matrix are extracted. As they represent angular changes between two frames, they can be considered as a representation of angular velocities when using small time intervals. The advantage of this approach is that the measurements are independent of the pose of the person with respect to the camera. This avoids some possible observation mismatches between training and testing due to the person being seen in a different pose with respect to the camera.

Furthermore, to fully characterize a rotation, only using the Euler angles (or visual velocities) is not sufficient. People may move their upper body back and forth, generating in this way

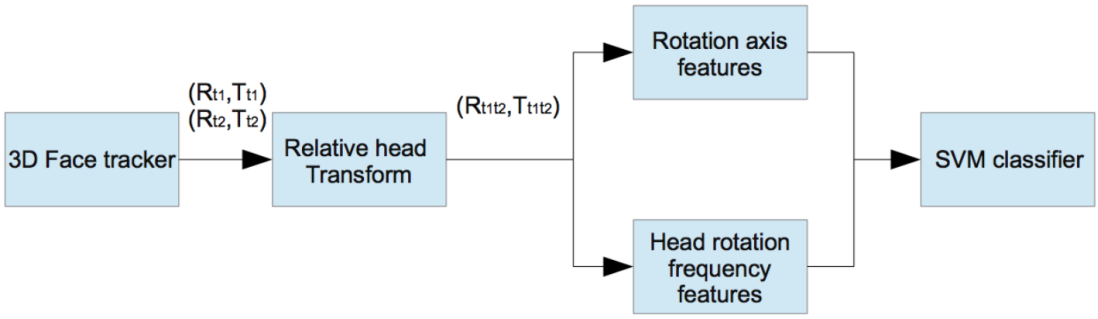


Figure 4.1: Overview of our nod detection system.

oscillatory pose angles. The main difference is that here the rotation axis might be located around the pelvis rather than around the neck. Thus, our system also proposes a feature related to rotation axis for classification. This feature could help distinguishing from which part of the body the rotation comes from, so that rotation movements not originating from the neck can be excluded.

4.2 Method Overview

The overall procedure of the approach is shown in Fig. 4.1. The head pose represented by a rotation and a translation of the face with respect to the camera coordinate system is first obtained from a 3D head tracker at each frame. Then the head rotation dynamic characterized by the head rotation $R_{t_1 t_2}$ and translation $T_{t_1 t_2}$ is computed within the head coordinate frame. In the next phase, two sets of features are extracted. First, similar to the work of Nguyen et al. [2012], our system applies a Fourier transform with Gaussian window to the rotation angles derived from $R_{t_1 t_2}$. Second, rotation axis features are also extracted from the relative translation and rotation. Finally, a SVM classifier is applied to all features. A more detailed description of each step is given below.

4.3 Relative Head Transformation

4.3.1 Head Pose Tracking

Instead of HeadFusion, we applied the method in [Funes-Mora and Odobez, 2012] to track the head pose because the work in this chapter is done before HeadFusion. Compared with HeadFusion, the tracking approach in [Funes-Mora and Odobez, 2012] only relies on 3D morphable model and depth information. Since detecting head nod in extreme scenarios (we used datasets of 2-party and 4-party daily conversations) is not our main focus, this simplified approach is fast and accurate enough.

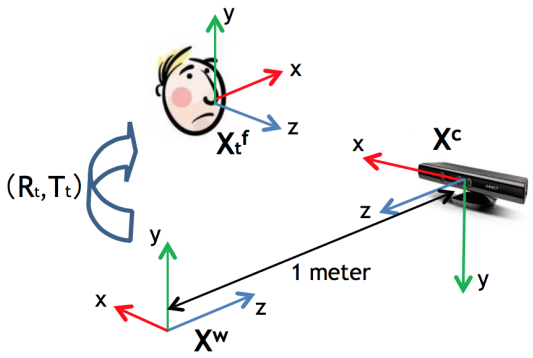


Figure 4.2: World and face coordinate system used by the head tracker.

4.3.2 Head Transformation with respect to Head Pose

Given a point P in the 3D space, we denote as $X^{cs}(P)$, the coordinates of this point in the coordinate system CS. In the face tracking system, there are two coordinate systems: the world coordinate system $X_t^w(P)$ which is fixed and located 1 meter away from the camera, and the face coordinate system $X_t^f(P)$, where t is the time since the face coordinate varies with time t. In the face coordinate system, the z axis is defined as the front direction of a person, whereas the x and y axis are defined as the side direction and the vertical direction respectively (see Fig. 4.2).

For a point P_t, the outputs of the tracker relate the face coordinate and camera coordinate. The corresponding transformation for every frame is represented by a 3×3 rotation matrix R_t and a translation vector T_t, defining:

$$X_t^w(P_t) = R_t X_t^f(P_t) + T_t\,. \tag{4.1}$$

We are interested in defining the transformation between the face coordinate systems at time $t_1 = t - m$ and time $t_2 = t$. Let us consider a face point P and let us denote by P_{t_1} and P_{t_2} its position in the 3D space at time t_1 and t_2. As the point is rigidly attached to the face, we have:

$$X_{t_1}^f(P_{t_1}) = X_{t_2}^f(P_{t_2})\,. \tag{4.2}$$

The point P_{t_2} at t_2 can be expressed in the face coordinate system at t_1 according to the transformation in Eq.4.3:

$$\begin{aligned} X_{t_1}^f(P_{t_2}) &= R_{t_1}^{-1}(X^w(P_{t_2}) - T_{t_1}) \\ &= R_{t_1}^{-1}(R_{t_2} X_{t_2}^f(P_{t_2}) + T_{t_2} - T_{t_1}) \\ &= R_{t_1}^{-1} R_{t_2} X_{t_2}^f(P_{t_2}) + R_{t_1}^{-1}(T_{t_2} - T_{t_1}) \\ &= R_{t_1 t_2} X_{t_1}^f(P_{t_1}) + T_{t_1 t_2} \end{aligned} \tag{4.3}$$

This equation represents the rigid rotation of a face point expressed in the coordinate system

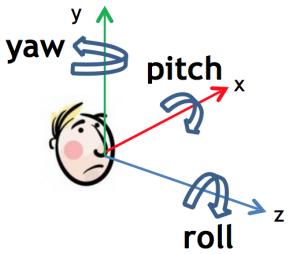


Figure 4.3: Euler angles defined in the head coordinate system.

of the face at t_1. Thus the relative transformation between t_1 and t_2 which is represented by the relative rotation matrix $R_{t_1 t_2}$ and relative translation $T_{t_1 t_2}$ is given by:

$$\begin{aligned} R_{t_1 t_2} &= R_{t_1}^{-1} R_{t_2}, \\ T_{t_1 t_2} &= R_{t_1}^{-1} (T_{t_2} - T_{t_1}) \end{aligned} \tag{4.4}$$

4.4 Feature Extraction and Classification

In this part, we first present our encoding into features of the transformation matrices $R_{t'-m,t'}$ defined for each time step t' of a short time window $[t-\Delta, t+\Delta]$ into features. Then we introduce the classification method used for head nod detection.

4.4.1 Rotation Frequency Features

At each time step t', we can extract from $R_{t'-m,t'}$ the three euler angles: roll, pitch, and yaw denoted by $(\alpha_{t'}, \beta_{t'}, \gamma_{t'})$ which are defined as the rotations around the z-, x- and y-axis (Fig. 4.3). We define $\alpha_{t-\Delta T:t+\Delta T}$ as the sequence of α observations within the temporal window $[t-\Delta T, t+\Delta T]$ (and similarly for β, γ), that is:

$$\alpha_{t-\Delta T:t+\Delta T} = [\alpha_{t-\Delta T}, \ldots, \alpha_{t+\Delta T}]. \tag{4.5}$$

In addition, we define a Gaussian window as:

$$\begin{aligned} W_{2\Delta T+1} &= [G(-\Delta T), \ldots, G(\Delta T)] \\ with\ G(n) &= e^{-\frac{1}{2}(\frac{n}{\sigma})^2}. \end{aligned} \tag{4.6}$$

In order to characterize the oscillatory nature of head nods around time t, we apply a Fourier transform along with a Gaussian window to these three angle series, leading to:

$$\begin{aligned} A_{-\Delta f:\Delta f} &= DFT(\alpha_{t-\Delta T:t+\Delta T} \cdot W_{2\Delta T+1}), \\ B_{-\Delta f:\Delta f} &= DFT(\beta_{t-\Delta T:t+\Delta T} \cdot W_{2\Delta T+1}), \\ \Gamma_{-\Delta f:\Delta f} &= DFT(\gamma_{t-\Delta T:t+\Delta T} \cdot W_{2\Delta T+1}) \end{aligned} \tag{4.7}$$

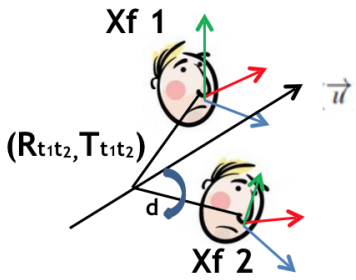


Figure 4.4: Head rotation around a fixed axis.

We then compute the norm of the output of the Fourier transform, which is defined as follows for A:

$$|A_k| = \sqrt{Re(A_k)^2 + Im(A_k)^2}. \tag{4.8}$$

Finally, we take the positive part of the normalized frequency spectrum to avoid redundancy from the three angle series and concatenate them to obtain the vector used as frequency features for the frame located at the center of the window:

$$\begin{aligned} f_m^{rot}(t) = [&|A_0|, \cdots, |A_{\Delta f}|, \\ &|B_0|, \cdots, |B_{\Delta f}|, \\ &|\Gamma_0|, \cdots, |\Gamma_{\Delta f}|] \end{aligned} \tag{4.9}$$

Remember that the m in f_m^{rot} refers to the frame gap between the two frames needed to compute the relative rotation $R_{t'-m,t'}$.

4.4.2 Rotation Axis Features

Only looking at the angles is not enough to describe a head movement. Indeed there can be some motions leading to similar angle changes like back and forth motion with the upper body, but which are not head nods. In this chapter, our goal is to capture that head rotations are done around an axis located near the neck. To do so, we compute the distance between the face and the rotation axis as additional but important feature characterizing the relative rotation.

Distance to Rotation Axis.

Let us consider a rigid transformation defined by the rotation R and translation T. By definition, the rotation axis is the set of points invariant to the rigid transformation, which can

therefore be obtained by solving the following equations:

$$\begin{aligned} X &= RX + T, \\ (I - R)X &= T \end{aligned} \tag{4.10}$$

There are three cases when solving for the above equation:

1. $R = I$ and $T \neq 0$. In this case, the set of points is empty. In practice, we will consider the distance to be at infinity and set the axis distance d to a large value.
2. $R = I$ and $T = 0$. In this case, the set of points is the 3D space. We consider that the distance is 0, and set $d = 0$.
3. $R \neq I$ and $T \neq 0$. In this case, the equation provides the rotation axis we are looking for.

Since we have a rotation around a fixed axis, the solution of Eq.4.10 is a line in the 3D space which means that $I - R$ is a singular matrix. To identify this axis, we can extract the direction of this line, and one point P^* from this line. For the latter one, we can use the least square solution as a particular solution of the equation. In our resolution, we use $(I - R) + \epsilon I$ instead of $I - R$ to stabilize the computation and avoid spurious values caused by noise (where ϵ is very small, we took 0.0001 in our calculation).

The null space of $I - R$ indicates the direction of the axis. In other words, to identify the axis direction, we can search for the unitary eigenvector $\vec{u}$ corresponding to the eigenvalue 1 of R, as every rotation matrix must have this eigenvalue (the other two being complex conjugates of each other), which can be found by solving:

$$R\vec{u} = \vec{u} \tag{4.11}$$

Then the distance between the origin O of the face coordinate system and the axis can be calculated as:

$$d = \|\overrightarrow{OP^*}\| \sin(\delta) \text{ with } \delta = \arccos\left(\frac{\overrightarrow{OP^*} \cdot \vec{u}}{\|\overrightarrow{OP^*}\| \|\vec{u}\|}\right).$$

Axis Features.

We can apply the above to the relative transformation defined by $R_{t'-m,t'}$ and $T_{t'-m,t'}$ and obtain the distance $d_{t'}$. Then, to summarize the axis information within the interval $[t-\Delta T, t+\Delta T]$, we take the maximum and average of the distance in the temporal window and define the axis features as:

$$f_m^{axis}(f) = [max(d_{t-\Delta T,t+\Delta T}), mean(d_{t-\Delta T,t+\Delta T})]$$

This feature can be used to eliminate false positives caused by body motion. Indeed, such motions like leaning forward and back, adjusting the sitting position, standing up and sitting down, may exhibit angular changes similar to nods. Our expectation is that the distance to the

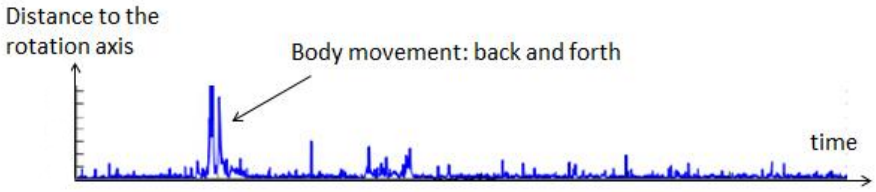


Figure 4.5: Distance of the relative axis of the relative rotation to the head frame origin.

axis will be able to distinguish them from nods since nods are rotation around the neck, and these body motions usually have their axis distance much farther, as illustrated in Fig. 4.5.

4.4.3 Classification

For the window $[t-\Delta, t+\Delta]$, we concatenate the Fourier features introduced in Section 4.4.1 as well as the maximum and average of the distance obtained in Section 4.4.2 into a single feature vector. Then, the system performs the classification of nods with this vector using a support vector machine (SVM). A support vector machine constructs a hyperplane which maximize its distance to the nearest training-data point of any class, since, in general, the larger the margin the lower the generalization error of the classifier. Some kernel functions can be used to implicitly project the data in a higher dimensional space where the date becomes more separable. The SVM classifier is applied at every frame. To filter out spurious detection, we applied a smoothing filter which eliminates detection events of very short duration (less than 7 frames).

4.5 Experimental Protocol

In this section, we will present the design of our experiments, including the data we used, the parameter setting, training and evaluation method.

4.5.1 Dataset

In our experiments, two datasets are used.

UBImpressed dataset. Acquired with a Kinect 2 sensor at 30fps, it consists of videos of job interviews. The camera is set about one meter away from the interlocutor and makes a little angle with the front direction (see Fig. 4.6, left, people are seen from below and the side). The dataset comprises 12 videos, each containing different people, for a total time duration of 60 minutes.

KTH-Idiap corpus [Oertel et al., 2014]. It features four people: one interviewer, and 3 interviewees who are applying for a funding grant. People are seated around a round table and each

Figure 4.6: UBImpressed sample (left) and view of the KTH-Idiap corpus setting (right).

Figure 4.7: Sample images of the KTH-Idiap corpus.

person was filmed by a Kinect 1 camera (See Fig. 4.6, right). The video frame rate is also 30 fps. Since the conversation happened around a round table, the participants tend to look at each other and turn their sides to the camera in the videos (See examples of people in Fig. 4.7). While full videos last around one hour, for the experiment we selected 5 minute excerpts from the videos of 9 different people.

4.5.2 Annotation

All the head nods were annotated manually. We annotated 13874 frames, for a total of 543 head nods in the two datasets (see Tab 4.1). The average duration of a nod is 25.5 frames (≈ 0.85s). Nods in KTH-Idiap are longer on average because there are more continuous multi-nods.

Since head nods might be difficult to define and different people hold different opinions towards ambiguous ones, we annotated two classes of nods: obvious and subtle, according to the human perceived amplitude and duration of the rotation movement. Around 50% of the nods were considered as obvious.

Table 4.1: Nod statistics for UBImpressed and KTH-Idiap dataset.

	#Nods	#Nod Frames	#Obvious Nods	Average Duration
UBImpressed	407	10252	201	25.2
KTH-Idiap	136	3622	83	26.6
Total	543	13874	284	25.5

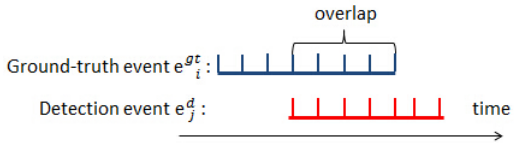


Figure 4.8: Nod matching.

4.5.3 Parameter Setting

The algorithm comprises several parameters. First, we looked at the parameter m, the time interval (measured as the number of the spacing frames) used to compute the relative rotation mentioned in section 4.3.2. In general, larger m might be more robust against the noise of the head tracker but may lead to detectors which are less sensitive to movement details. In our experiment we tried with $m = 1, 3, 5, 7$.

Another parameter is the size of the Gaussian window $2\Delta T + 1$. In our experiments, we chose 31 frames (about 1 second) so $\Delta T = 15$. Note that the resulting window duration is larger than the duration of 90% of the annotated nods.

Apart from that, we chose the LIBSVM package as the SVM library tool. SVM parameters were chosen via 5-fold cross validation within the training set with a grid search. Note that in all cases, the feature vectors were z-normalized (i.e. the mean was subtracted and the result was divided by the standard deviation).

4.5.4 Performance Measurement

The performance of the head nod detector is measured at the frame and event levels. At the frame level, we used the standard precision, recall and F-score measures. At the event level, we first need to match recognized events with the ground truth. To do so, suppose that there is a nod event e_i^{gt} (ground truth) happening in the time interval $I_i^{gt} = [t_{i,s}^{gt}, t_{i,t}^{gt}]$ and a detected nod event e_j^d in $I_j^d = [t_{j,s}^d, t_{j,t}^d]$ (see Fig. 4.8). Then the event matching precision, recall, and F-score between e_j^{gt} and e_i^d are defined as:

$$P_{i,j}^{ov} = \frac{|I_i^{gt} \cap I_i^d|}{|I_i^{gt}|}, R_{i,j}^{ov} = \frac{|I_i^{gt} \cap I_i^d|}{|I_i^d|}, F_{i,j}^{ov} = \frac{2P_{i,j}^{ov} \cdot R_{i,j}^{ov}}{P_{i,j}^{ov} + R_{i,j}^{ov}}.$$

The events are said to match if their F-score is above a threshold (in that case e_i^{gt} is considered detected and e_i^d is considered correct). One difficulty arises in the case of long lasting multiple nods, which are difficult to annotate (as a single long nod, as separate short ones). In order to account for this situation, we set the threshold as 0.1

Then, given the matched events, we can compute the event-level precision, recall and F-score

as follow:

$$
\begin{aligned}
P_{event} &= \frac{\#\left\{e_j^d | \exists i, F_{i,j}^{ov} > threshold\right\}}{\#e^d}, \\
R_{event} &= \frac{\#\left\{e_i^{gt} | \exists j, F_{i,j}^{ov} > threshold\right\}}{\#e^{gt}}, \\
F_{event} &= \frac{2P_{event} \cdot R_{event}}{P_{event} + R_{event}}.
\end{aligned}
\tag{4.12}
$$

4.5.5 Model Setups

We trained 3 different support vector machines, with 3 different feature sets:

- **Baseline**: this corresponds to the work in [Nakamura et al., 2013; Wei et al., 2013; Morency et al., 2007], where the Fourier transform outputs of sequences of Euler angle differences computed using the pose matrix defined with respect to the camera frame are used as feature.
- **Relative rotation (RelRot)**: In this case, the Fourier features $f_m(t) = f_m^{rot}(t)$ of the angle extracted from the relative rotation matrix are used, as shown in section 4.4.1.
- **Relative rotation + axis distance (RelRot-AxisDist)**: in addition to the rotation features, the average and maximum of the distance to the rotation axis is used. That is, $f_m(t) = [f_m^{rot}(t), f_m^{axis}(t)]$.

4.5.6 Implementation Details

As very subtle head nods can be similar to non-nod movements during speaking, only obvious nods were used as training samples[1]. Furthermore, to avoid introducing noise in the learning stage, we defined as transition frames 7 frames before and after the onset and offset frames of a nod, and did not use them as training samples (either as negative or positive samples). Note that by using only the central part of the nods as training data, we can guarantee that in general the main part (at least three quarters) of the Gaussian window used to compute the frequency overlaps the nods (See Fig. 4.9).

Negative samples were chosen randomly, and more negative samples were chosen than positive ones since the space of negative gestures is larger than the space of positive ones. At the end we had 3100 positive samples and 10000 negative samples in the UBImpressed data.

4.6 Results

Experiments are conducted on the two datasets separately. In section 4.6.1 we present the results obtained on the UBImpressed dataset. Then in section 4.6.2, we apply the trained

[1] Note that this only concerns training. Subtle nods were kept in the test set for evaluation.

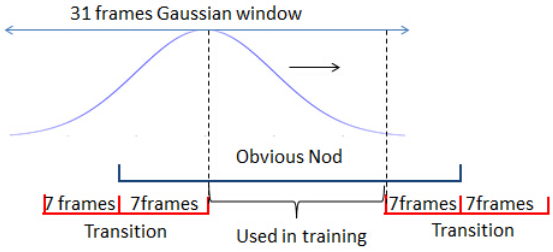


Figure 4.9: Transition frames around nod start and ends were not used for training

Table 4.2: Results of UBImpressed data.

	EventLevel			*FrameLevel*		
	Precision	Recall	F-score	Precision	Recall	F-score
linear SVM, m = 3						
Baseline	0.8	0.83	0.81	0.8	0.73	0.76
RelRot	0.81	0.83	0.82	0.8	0.72	0.76
RelRot-AxisDist	0.81	0.83	0.82	0.78	0.75	0.76
RBF SVM, m = 3						
Baseline	0.84	0.8	0.82	0.84	0.68	0.75
RelRot	0.85	0.81	0.83	0.84	0.68	0.75
RelRot-AxisDist	0.87	0.8	0.84	0.83	0.7	0.76
linear SVM, m = 5						
Baseline	0.81	0.83	0.82	0.82	0.74	0.78
RelRot	0.84	0.83	0.84	0.82	0.75	0.78
RelRot-AxisDist	0.82	0.85	0.83	0.8	0.77	0.78
RBF SVM, m = 5						
Baseline	0.86	0.79	0.82	0.86	0.7	0.77
RelRot	0.86	0.8	0.83	0.87	0.71	0.78
RelRot-AxisDist	0.87	0.82	0.84	0.86	0.73	0.79

classifier on KTH-Idiap dataset to test the generalization performance.

4.6.1 UBImpressed Data

A leave-one-person cross validation experiment was performed among the UBImpressed dataset. That is, the SVM classifier was trained with samples from 11 videos and tested on the last one by applying nod detector to the entire video. All videos make turns to be the test sample. We used both radial basis function kernel and linear kernel for $m = 1, 3, 5, 7$.

Overall results. Tab. 4.2 reports the results. In general, we obtain a F-score of 0.83 at the event level, and of 0.76 at the frame level. In this latter case, we can notice the higher precision and lower recall, which might be due to the use of only obvious nods during learning, shown in section 4.5.6. Overall, our results are quite high, when considering the subtleness of most of

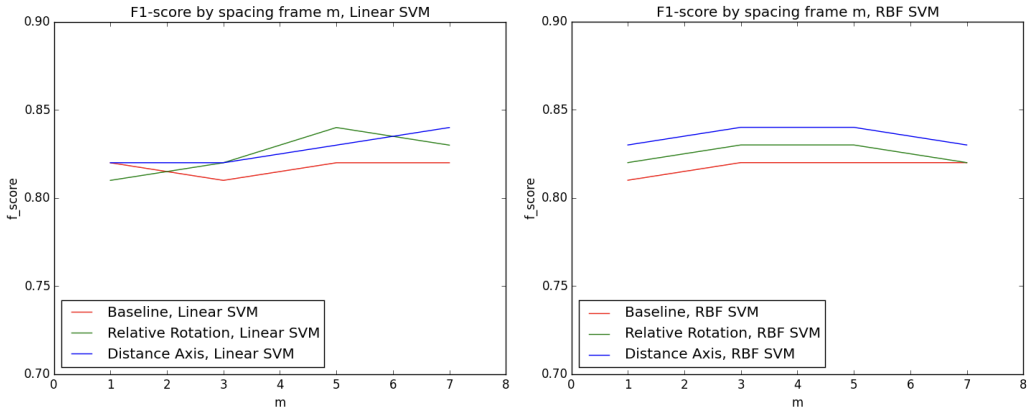


Figure 4.10: Event-level results in function of the time interval m. Left: linear SVM results; Right: RBF SVM results.

the examples.

Influence of time interval m. Tab. 4.2 and Fig. 4.10 report the impact of changing the time interval used to compute the relative rotations (and hence, approximation of angular speed). We can notice that in general, the best results are obtained with $m = 3$ and $m = 5$, while results with $m = 1$ are lower (affected by potential tracker instability). With $m = 7$, results are more contrasted. These results are confirmed by the precision-recall curve measured at frame-level, shown at the top of Fig. 4.11.

Model comparison. Tab. 4.2, Fig. 4.10 and Fig. 4.11 provide a comparison of the different feature vectors. As can be seen, the use of relative rotation features and axis distance produce slightly better results. Indeed, in the configuration of UBImpressed (see Fig. 4.6), all people are seen from the same viewpoint, and look towards the job interviewer, so we have very similar head pose. Since we train and test from the same dataset, the invariance does not bring much.

To validate that the relative rotation is more robust, we generate a series of synthetic data by systematically rotating the head, to simulate a change of viewpoint. To do so, we transformed the sequence of head pose R_t into $R'_t = R^{vp} R_t$. R^{vp} can simulate a change of pitch and roll (which can be due to being seen from below/above or with an in-plane rotated camera). Thus we trained a model from the original data and tested it on the held out video with the modified viewpoint. The results are shown in Fig. 4.12. While the relative rotation are by definition not affected by such a change, the sequence of Euler angles measured in camera are affected, with a performance reduction of 10% at 20° viewpoint change. Such behaviour is also observed on the KTH data.

As motivated earlier, the distance to rotation axis could be a useful feature to filter out false alarms due to body movements. However, we find this feature does not alter the overall results much, sometimes even degrade the precision. It might be due to two reasons: i) the false alarms of head nod can involve complex head rotations around multiple axes rather than

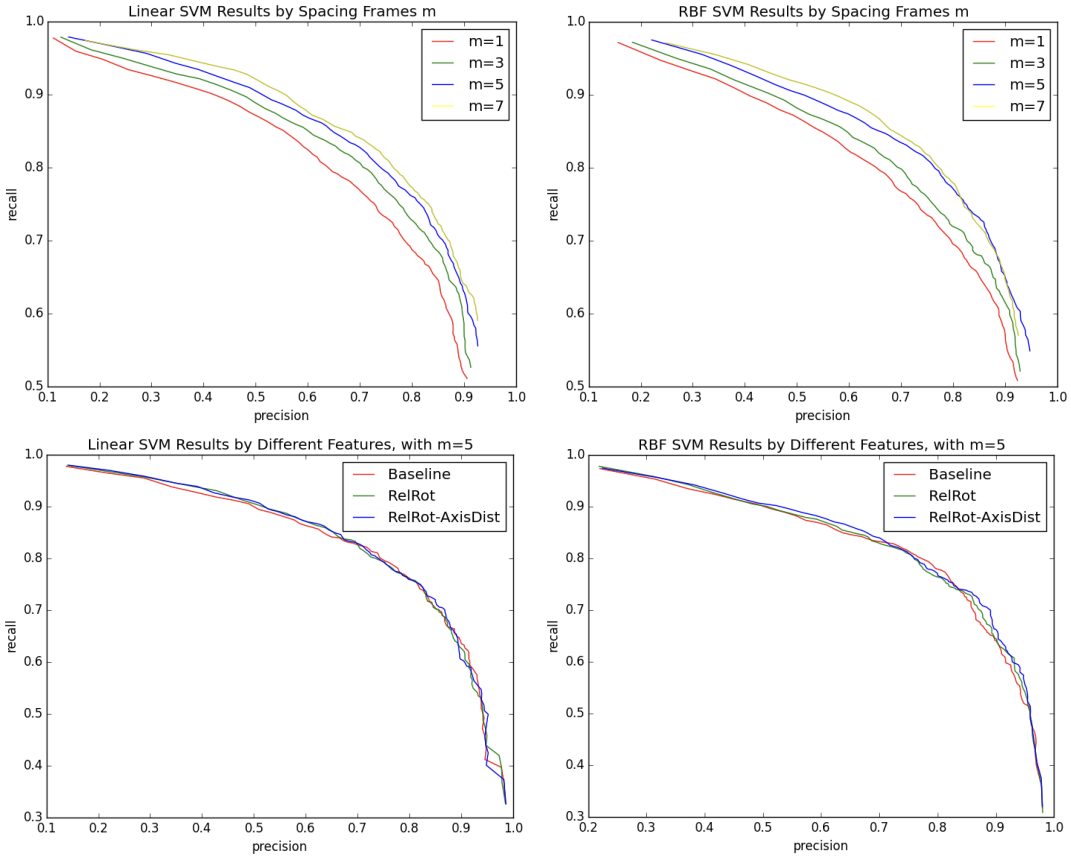


Figure 4.11: Frame-level results in function of the time interval and features. Top Left: linear SVM results; Top Right: RBF SVM results; Bottom Left: linear SVM results ($m = 5$); Bottom Right: RBF SVM results ($m = 5$).

a single axis on the body, thus the assumption of this feature might not be solid enough; ii) the false alarms of the nod behaviour might be seldom in our dataset where the participants were seated and have less degree of freedom. In both cases, the distance feature can become ineffective.

Finally, we show in Fig. 4.13 the weights of the linear SVM. It can be seen that as expected, the filter reacts to rotation around the pitch in the 1-4Hz range, while is negatively affected by low frequency gestures around the yaw and roll rotations axis, which reflects the fact that real nods should only involve pitch, and not be a composite of rotations. In addition, we can notice that the rotation axis distance feature (esp. the average one) also negatively vote against the nod detection, as we could expect.

Error analysis. Qualitatively, most false positive errors are due to single stroke lowering or raising head gesture (with a small overshoot/oscillation at the end due to momentum control). False negatives come from very light nods or nods accompanied by other gestures, esp. during speaking periods. Note that during speaking time, some very subtle head gestures are difficult

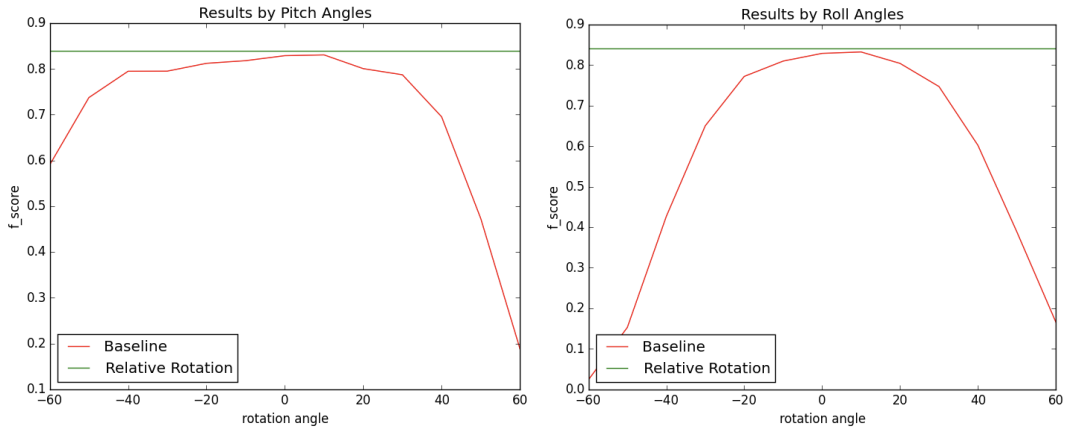


Figure 4.12: Results obtained by simulating on the test data a view-point change. Left: change in pitch (looking from above/bottom). Right: change in roll (looking with more in-plane rotation).

to label as nod or not, due to the presence of other head related motions/activities. Thus, some annotations inaccuracy and the trembling of the head pose tracker could influence the result.

4.6.2 KTH-Idiap Data

We first evaluated the generalization capabilities of our model and used the nod models trained on UBImpressed data and tested them on the KTH-Idiap sequences. Results are reported in Tab. 4.3. Two main remarks can be made. First, results are lower than on the UBImpressed data overall (F-score of 0.72 vs 0.84 for events), which can due to multiple factors. Most importantly, as people are more distant to the sensor, and are less facing the camera, tracking (remember that our tracker only relies on depth information) is more difficult and results in noisier head pose sequences. Furthermore, as there are four people, people behave more often as observers, producing some very light nods, often from a side view, which makes their recognition very challenging. Secondly, we can notice that the use of the relative head rotation features results in better performance (0.72 vs 0.68 for events), demonstrating their greater invariance to viewpoint and pose changes.

Finally, we also trained the classifiers on KTH-Idiap dataset using a one-person leave out scheme. Results are shown in Tab. 4.4. Surprisingly, results are not necessarily higher than with the model trained on UBImpressed data, (0.68 vs 0.72 at the event level). This might be due to the use of noisier head pose tracking features during training, which affects the recognition behaviour (see the drop in precision). Nevertheless, note that the proposed features still usually perform better than the baseline, especially at the event level.

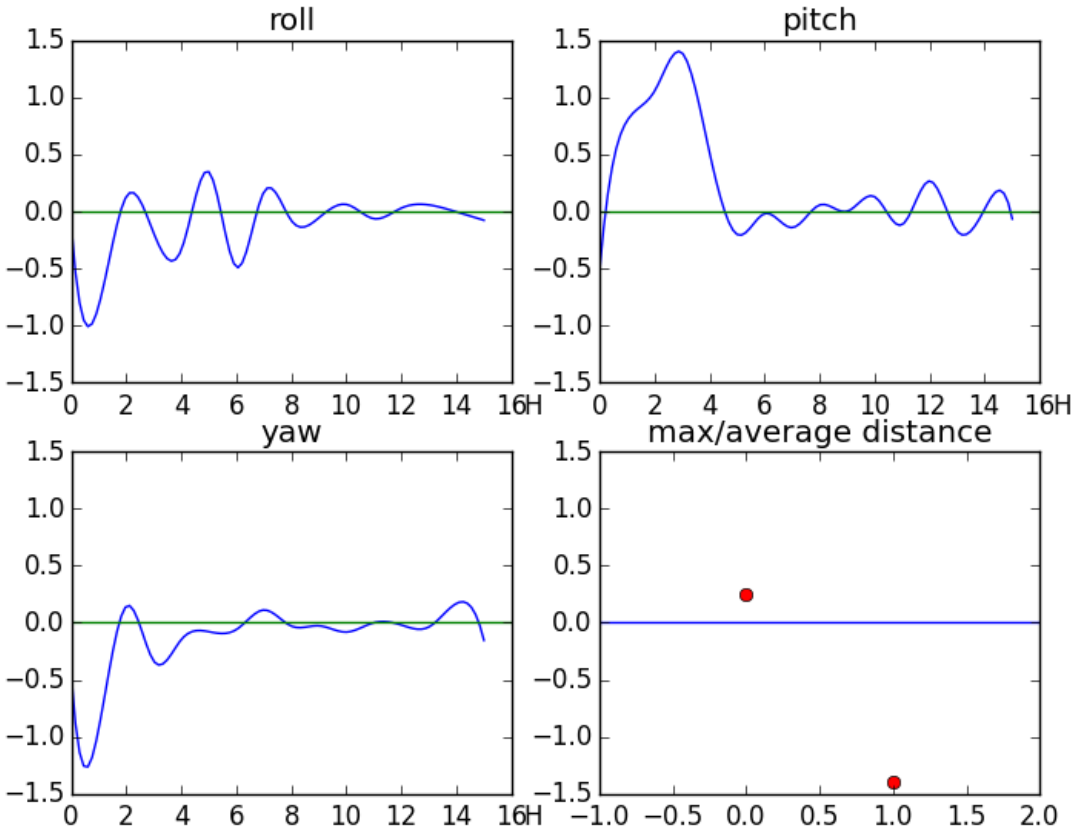


Figure 4.13: Weights for linear SVM, $m = 3$.

4.7 Conclusion

In this chapter, we developed a head nod detection system. The system exploits the 3D oscillatory characteristics of nods by relying on the Euler angles extracted from the relative rotation matrix expressed in the camera frame and on the distance of the rotation axis to the face origin. Compared to previous approaches, the method improves the detection and provides accurate results, even in the case of subtle nods. It is possible to extend this method to other head gestures like head shake. Future work can consist of further exploring the role of the rotation axis for recognition, e.g. by testing the system with standing people. In addition, investigation on the exploitation of a temporal model (e.g. CRF) as well as multiple instance learning would be helpful.

Table 4.3: Results of KTH-Idiap data, with nod detector trained on UBImpressed data.

	EventLevel			*FrameLevel*		
	Precision	Recall	F-score	Precision	Recall	F-score
linear SVM, $m=3$						
Baseline	0.73	0.63	0.68	0.72	0.44	0.55
RelRot	0.75	0.69	0.72	0.78	0.47	0.59
RelRot-AxisDist	0.74	0.69	0.72	0.79	0.48	0.6
RBF SVM, $m=3$						
Baseline	0.83	0.54	0.66	0.86	0.38	0.53
RelRot	0.83	0.62	0.71	0.87	0.42	0.57
RelRot-AxisDist	0.81	0.62	0.7	0.86	0.42	0.57
linear SVM, $m=5$						
Baseline	0.74	0.63	0.68	0.76	0.46	0.58
RelRot	0.75	0.7	0.72	0.81	0.5	0.61
RelRot-AxisDist	0.73	0.69	0.71	0.82	0.5	0.62
RBF SVM, $m=5$						
Baseline	0.81	0.57	0.67	0.86	0.43	0.57
RelRot	0.83	0.61	0.7	0.87	0.44	0.59
RelRot-AxisDist	0.83	0.63	0.72	0.87	0.45	0.59

Table 4.4: Results of KTH-Idiap data, trained on KTH-Idiap data.

	EventLevel			*FrameLevel*		
	Precision	Recall	F-score	Precision	Recall	F-score
linear SVM, $m=3$						
Baseline	0.54	0.8	0.65	0.6	0.59	0.59
RelRot	0.54	0.79	0.64	0.59	0.58	0.59
RelRot-AxisDist	0.54	0.79	0.64	0.59	0.58	0.59
RBF SVM, $m=3$						
Baseline	0.57	0.77	0.65	0.57	0.57	0.57
RelRot	0.6	0.8	0.68	0.59	0.58	0.59
RelRot-AxisDist	0.6	0.8	0.68	0.59	0.58	0.59
linear SVM, $m=5$						
Baseline	0.53	0.79	0.63	0.6	0.61	0.61
RelRot	0.57	0.8	0.66	0.61	0.61	0.61
RelRot-AxisDist	0.57	0.8	0.66	0.61	0.61	0.61
RBF SVM, $m=5$						
Baseline	0.55	0.77	0.64	0.57	0.6	0.58
RelRot	0.6	0.79	0.68	0.59	0.62	0.61
RelRot-AxisDist	0.6	0.79	0.68	0.59	0.62	0.61

5 Multitask Learning for Gaze Estimation

In this chapter, we present a 3D gaze estimation approach (retrieving the 3D Line of Sight which is expressed by a pitch angle and a yaw angle) based on multi-task learning. To train the network, we rely on both synthetic data and real data where the two datasets provide labels of different tasks. Besides, we also propose a Constrained Landmark-Gaze Model (CLGM) which models the eye landmarks and gaze jointly.

The rest of this chapter is organized as follows: we motivate our main idea and list the contributions in Chapter. 5.1; Some background on multi-task learning is given in Chapter. 5.2; The correlation between eye landmarks and gaze is analyzed in Chapter. 5.3; Chapter. 5.4-5.6 presents the method overview, the Constrained Landmark-Gaze Model and our multi-task learning network respectively; The experiment protocol and results are demonstrated in Chapter 5.7 and 5.8 respectively; Finally, we make a summary of our approach in Chapter. 5.9.

5.1 Motivation and Contributions

As mentioned in Chapter. 2, although progress has been reported by deep learning based approaches, direct regression of gaze still suffers from limitations such as the lack of data and inaccurate eye cropping. To address these issues, we propose an end-to-end trainable deep multitask framework based on a Constrained Landmark-Gaze Model, with the following properties.

First, we address eye landmark (including iris center) detection and gaze estimation jointly. Indeed, since gaze values are strongly correlated with eye landmark locations, we hypothesize that modelling eye landmark detection (which is an explicit visual task) as an auxiliary task can ease the learning of a predictive model of the more abstract gaze information. To the best of our knowledge, this is the first time that multitask learning is applied to gaze estimation. Since there is no existing large scale dataset with annotated eye landmarks (>10 landmarks) and gaze, we rely on a synthetic dataset for the learning of the auxiliary task in this work. Note that we only use the landmark annotations from the synthetic data because of the different

gaze setting of synthetic data. The use of synthetic data also expands the amount of training data to some extent.

Second, instead of predicting eye landmarks and gaze in two network branches as in usual deep multitask learning, we build a Constrained Landmark-Gaze Model (CLGM) modelling the joint variation of eye landmark location and gaze direction, which bridges the two tasks in a closer and more explicit way.

Third, we make our approach more robust to scale, translation and even head pose variations by relying on a deterministic decoder. More precisely, the network learns two sets of parameters, which are the coefficients of the CLGM model, and the scale and translation parameters defining the eye region. Using these parameters and the head pose, the decoder deterministically predicts the eye landmark locations and gaze via the CLGM. Note however that while all parameters account for defining the landmark positions, only the CLGM coefficients and the head pose are used for gaze prediction. Thus, gaze estimation is decoupled from irrelevant variations in scale and translation and geometrically modelled within the head pose frame.

Finally, note that while currently landmark detection is used as a secondary task, it could be used as a primary task as well to extract the features (eye corners, iris center) requested by a geometrical eye gaze model, which can potentially be more accurate. In particular, the CLGM could help predicting iris location even when the eyes are not fully open (see Fig. 5.7 for examples).

Thus, in summary, our contributions are as follows:

- A Constrained Landmark-Gaze Model modelling the joint variation of eye landmarks and gaze;
- Gaze estimation robust to translation, scale and head pose achieved by a CLGM based decoder;
- An end-to-end trainable deep multitask learning framework for gaze estimation with the help of CLGM and synthetic data.

5.2 Background on Multi-task Learning

Multitask learning aims to improve the overall performance of one or each task by providing implicit data augmentation or regularizations [Ruder, 2017]. Due to the flexibility of network architectures, a number of works have been proposed on deep multitask learning. The classical implementation is to share parameters in shallow layers and arrange task-specific branches in deeper layers. Many of the representative works are face-related research [Ranjan et al., 2016, 2017; Wang et al., 2017; Zhang et al., 2014; Yim et al., 2015] since there are plenty of datasets with rich annotations in this area and the face attributes are also well correlated. Some other works, however, attempted to propose novel multitask learning architectures which could generalize well on other tasks. For example, the Cross-stitch Network [Misra et al.,

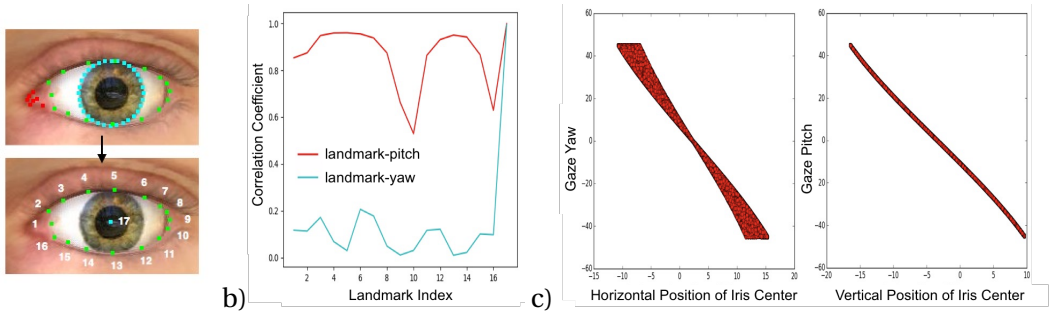


Figure 5.1: Correlation between the eye landmark positions and gaze values, computed from the UnityEyes dataset. (a) Selected landmarks (bottom) from the UnityEyes landmark set (top). (b) Correlation coefficients between the landmark horizontal or vertical positions and the gaze yaw or pitch angles. (c) Joint distribution map of the horizontal or vertical positions of the iris center and of the yaw or pitch gaze angles.

2016] designed a cross-stitch unit to leverage the activations from multiple models thus the parameters are shared softly. However, the network architecture and the placing of cross-stitch are still manually determined. Instead of hand designing the multitask learning architecture, Lu et al. [2016] proposed to dynamically create network branches for tasks during training so fully adaptive feature sharing is achieved. Nevertheless, their approach did not model the interactions between tasks.

5.3 Correlation of Eye Landmarks and Gaze

Before introducing our method, we first study the correlation existing between the gaze and eye landmarks. We used the synthetic database UnityEyes [Wood et al., 2016b] for correlation analysis since this database provides rich and accurate information regarding the landmark and gaze values. As this dataset relies on a synthetic yet realistic model of the eye ball and shape, and since this database has been used for the training of gaze estimators which have achieved very high performance on real datasets [Wood et al., 2016b], we expect this correlation analysis to be rather accurate. In any case, in Section 5.6.3, we show how we can account for the discrepancy between the synthetic model and real data.

Landmark set. The UnityEyes annotates three types of eye landmarks, the caruncle landmarks, the eyelid landmarks and the iris landmarks, as shown in the first row of Fig. 5.1a. Considering that relying on many landmarks will not help improving the robustness and accuracy but simply increase the complexity of the method, we only selected a subset $\mathscr{I}$ of the available landmark instead. It contains 16 landmarks from the eyelid, and the iris center which is estimated from the iris contour landmarks. This is illustrated in the second row of Fig. 5.1a.

Landmark alignment. We generated 50,000 UnityEyes samples with frontal head pose and a gaze value uniformly sampled within the [-45°, 45°] range for both pitch and yaw. All the

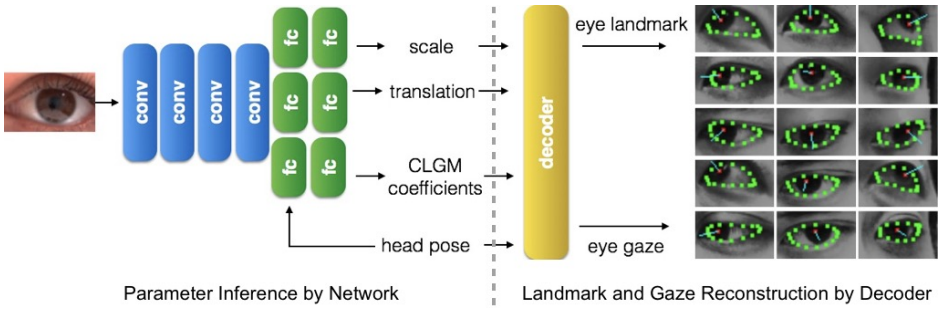


Figure 5.2: Framework of the proposed method.

samples are aligned on a global center point c_l.

Correlation analysis. We assign the landmark indices as shown in Fig. 5.1a. Then we compute the correlation coefficient between each landmark and gaze coordinates. More precisely, two correlation coefficients are computed: the gaze yaw - horizontal landmark position and the gaze pitch - vertical landmark position. They are displayed in Fig. 5.1b.

The following comments can be made. First, the position of the iris center (landmark 17) is strongly correlated with gaze, as expected. The correlation coefficient between the horizontal (respectively vertical) position of the iris center and the gaze yaw (respectively pitch) is close to 1. Furthermore, the joint data distribution between the iris center and gaze displayed in Fig. 5.1c indicates that they seem to follow a linear relationship, especially in the pitch direction. Second, the gaze pitch is also highly correlated with other landmarks (red curve in Fig. 5.1b). This reflects that looking up or looking down requires some eyelid movement which are thus quite indicative of the gaze pitch. Third, the gaze yaw is only weakly correlated with eyelid landmarks, which means that looking to the left or right is mainly conducted by iris movements.

In summary, we find that the eye landmarks are correlated with the gaze and therefore they can provide strong support cues for estimating gaze.

5.4 Method Overview

The proposed framework is shown in Fig. 5.2. It consists of two parts. The first part is a neural network which takes an eye image as input and regresses two sets of parameters: the coefficients of our joint CLGM landmarks and gaze model, and the scale and translation defining the eye region. The second part is a deterministic decoder. Based on the Constrained Landmark-Gaze model, it reconstructs the eye landmark positions and gaze with the two sets of parameters and the head pose. Note that in the reconstruction, the eye landmarks are computed using all parameters while the gaze is only determined using the CLGM coefficients and the head pose. An end-to-end training of the network is performed by combining the losses on landmark localization and gaze estimation. In our approach, we assume that the head pose has been obtained in advance. Below, we provide more details about the different

parts of the model.

5.5 Constrained Landmark-Gaze Model

As with the 3D Morphable Model [Paysan et al., 2009] or the Constrained Local Model [Cristinacce and Cootes, 2006] for faces, the eye shape can also be modelled statistically. Concretely, an eye shape can be decomposed as a weighted linear combination of a mean shape and a series of deformation bases according to:

$$\mathbf{v}_l(\alpha) = \mu^l + \sum_j \alpha_j \lambda_j^l \mathbf{b}_j^l, \tag{5.1}$$

where μ^l is the mean eye shape and λ_j^l represents the eigenvalue of the j^{th} linear deformation basis $\mathbf{b}_j^l$. The coefficients α denote the variation parameters determining eye shape while the superscript l means landmark.

As demonstrated in the previous section, the eye landmark positions are correlated with the gaze directions. In addition, we can safely assume that the landmark positions are also correlated. Therefore, we propose the Constrained Landmark-Gaze Model to explicitly model the joint variation of eye landmarks and gaze.

Concretely, we first extract the set of landmarks $\mathscr{I}$ from the N_s UnityEyes samples and align them with the global eye center. Denoting by $\mathbf{l}_{k,i} = (\mathbf{l}_{k,i}^x, \mathbf{l}_{k,i}^y)$, the horizontal and vertical positions of the i^{th} landmark of the k^{th} UnityEyes sample, and by $(\mathbf{g}_k^\phi, \mathbf{g}_k^\theta)$ the gaze pitch and yaw of the same sample, we can define the 1-D landmark-gaze array:

$$[\mathbf{l}_{k,1}^y, \cdots, \mathbf{l}_{k,N_l}^y, \mathbf{l}_{k,1}^x, \cdots, \mathbf{l}_{k,N_l}^x, \mathbf{g}_k^\phi, \mathbf{g}_k^\theta] \tag{5.2}$$

where N_l denotes the number of landmarks ($N_l = 17$), and the superscripts y, x, ϕ, θ represent the vertical position, horizontal position, pitch angle and yaw angle, respectively. This landmark-gaze vector has $2N_l + 2$ elements.

We then stack the vector of each sample into a matrix $\mathbf{M}$ of dimension $N_s \times (2N_l + 2)$, from which the linear bases $\mathbf{b}_j^{lg}$ representing the joint variation of eye landmark locations and gaze directions are derived through Principal Component Analysis (PCA). Thus, the eye shape and gaze of any eye sample can be modelled as:

$$\mathbf{v}_{lg}(\alpha) = \mu^{lg} + \sum_{j=1}^{2N_l+2} \lambda_j^{lg} \alpha_j \mathbf{b}_j^{lg} \tag{5.3}$$

where the superscript lg denotes the joint modelling of landmark and gaze. The definition of other symbols are similar to those in Eq. 5.1. Note that the resulting vector $\mathbf{v}_{lg}(\alpha)$ contains both the eye shape and gaze information.

In Eq. 5.3, the only variable is the vector of coefficients α. With a suitable learning algorithm,

α can be determined to generate an accurate eye shape and gaze.

Note that the current CLGM version only reflects the correlation between gaze and eye landmarks of synthetic data. Its adaptation to real data would be discussed in Section. 5.6.3.

5.6 Joint Gaze and Landmark Inference Network

We use a deep convolutional neural network to jointly infer the gaze and landmark locations, as illustrated in Fig. 5.2. It comprises two parts: an encoder network inferring the coefficient α of the model in Eq. 5.3, as well as other geometric parameters, and a decoder computing the actual landmark positions in the image and the gaze directions. The specific architecture for the encoder is described in Section 5.6.4. Below, we detail the decoder component and the loss used to train the network.

5.6.1 Geometric Decoder

We recall that the vector $\mathbf{v}_{lg}(\alpha)$ from the CLGM model only provides the aligned landmark positions. Thus, to model the real landmark positions in the cropped eye images, the head pose, the scale and the translation of the eye should be taken into account. In our framework, the scale s and translation $\mathbf{t}$ are inferred explicitly by the network, while the head pose is assumed to have already been estimated (see Fig. 5.2).

Given the head pose $\mathbf{p}$ and the inferred parameters α, s and $\mathbf{t}$ from the network, a decoder is designed to compute the eye landmark locations and gaze direction. Concretely, the decoder first uses α to compute the aligned eye shape and gaze according to Eq. 5.3. Then the aligned eye shape is further transformed with the head pose rotation matrix $\mathbf{R}(\mathbf{h})$, the scale s and the translation $\mathbf{t}$ to reconstruct the eye landmark positions in the input image:

$$\begin{bmatrix} \mathbf{l}_p^x \\ \mathbf{l}_p^y \end{bmatrix} = s \cdot \mathbf{Pr} \cdot \mathbf{R}(\mathbf{h}) \cdot \begin{bmatrix} \mathbf{v}_{lg}^x(\alpha) \\ \mathbf{v}_{lg}^y(\alpha) \\ 0 \end{bmatrix} + \mathbf{t} \tag{5.4}$$

$$\begin{bmatrix} cos(\mathbf{g}_p^\phi) sin(\mathbf{g}_p^\theta) \\ -sin(\mathbf{g}_p^\phi) \\ cos(\mathbf{g}_p^\phi) cos(\mathbf{g}_p^\theta) \end{bmatrix} = \mathbf{R}(\mathbf{h}) \cdot \begin{bmatrix} cos(\mathbf{v}_{lg}^\phi(\alpha)) sin(\mathbf{v}_{lg}^\theta(\alpha)) \\ -sin(\mathbf{v}_{lg}^\phi(\alpha)) \\ cos(\mathbf{v}_{lg}^\phi(\alpha)) cos(\mathbf{v}_{lg}^\theta(\alpha)) \end{bmatrix} \tag{5.5}$$

where $\mathbf{l}_p$ and $\mathbf{g}_p$ denote the predicted eye landmark positions and gaze respectively, and $\mathbf{Pr}$ is the projection matrix from 3D to 2D. From the equations above, note that the eye landmark positions are determined by all parameters while the gaze angles are only determined by the coefficient α and the head pose. Thus gaze estimation is geometrically coupled with the head pose as it should be, but is decoupled from the eye scale and translation.

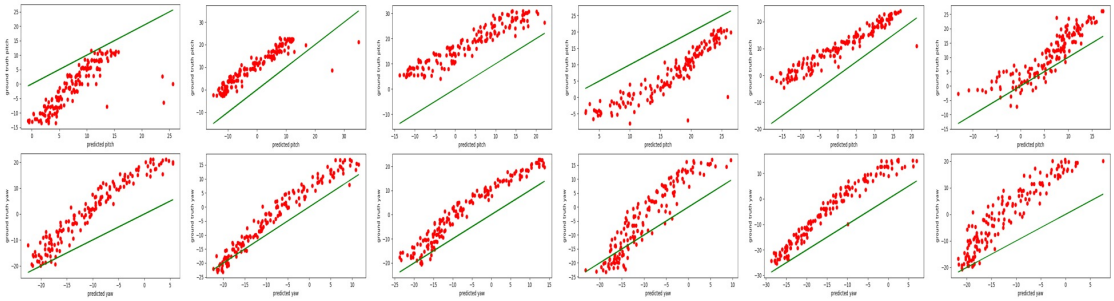


Figure 5.3: Gaze bias between prediction and ground truth. 1^{st} row: pitch angle. 2^{nd} row: yaw angle. Green line: identity mapping

5.6.2 Multi-task Loss.

To train the network, we define loss on both the predicted eye landmark positions and on the gaze according to

$$L(I) = w_l||\mathbf{l}_p - \mathbf{l}_g||_2 + w_g||\mathbf{g}_p - \mathbf{g}_g||_1 \tag{5.6}$$

where $\mathbf{l}_g$ and $\mathbf{g}_g$ represent the ground truth in the image I for the landmark positions and gaze respectively, and w_l and w_g denote the weights for the landmark loss and gaze loss respectively. Note from Eq. 5.6 that we do not provide any ground truth for scale or translation during training (or to α), since the network automatically learns how to predict them from the landmark loss.

5.6.3 CLGM Revisited

The CLGM model only reflects the correlation between gaze and eye landmarks of synthetic data. To account for real people and real images and obtain a more accurate CLGM model, we perform an evaluation guided correction of the CLGM model. The main idea is to evaluate how our gaze prediction approach trained only with synthetic data (for both the CLGM model and the network model) performs on target real data. Then, by comparing the gaze predictions with the actual gaze data for a subject, we can estimate a gaze correction model mapping the prediction to the real ones. Such a parametric model can then be exploited on the UnityEyes data to correct the gaze values associated with a given eye landmarks configuration. A corrected CLGM model can then be obtained from the new data, and will implicitly model the joint variations of eye landmarks on the real data with the actual gaze on real data.

More concretely, we proceed as follows. We first train a gaze estimator (and landmark detector at the same time) with the proposed framework using only the UnityEyes synthetic data. Then the synthetic trained estimator is applied on a target database (UTMultiview, Eyediap) comprising N_{sub} subjects. For each subject, we can obtain gaze prediction/ground truth pairs, as illustrated in Fig. 5.3. According to these plots, we found a linear model (different for each subject) can be fitted between the prediction and the ground truth. In other words, the gaze

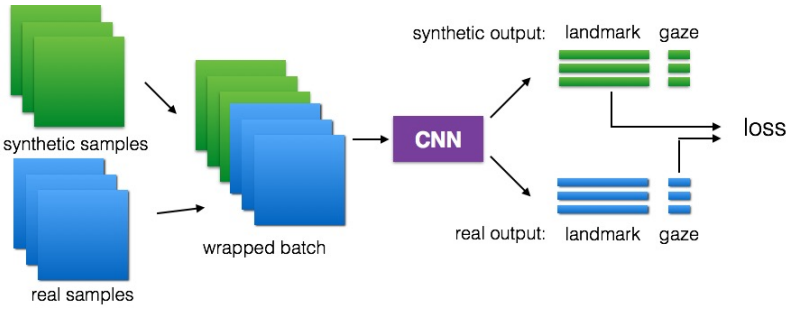


Figure 5.4: Training combining synthetic and real data.

predicted by the synthetic model is biased with respect to the real one but can be corrected by applying a linear model. Thus, to obtain a CLGM model linked to real people, for each subject j we fit two linear models f_j^{ϕ} and f_j^{θ} for the pitch and yaw prediction. Then, using the UnityEyes images, we construct a matrix $\mathbf{M}_j$ similar to the $\mathbf{M}$ matrix in Section 5.5, but stacking now the following landmark-gaze vectors instead of those in Eq. 5.2:

$$[\mathbf{l}_{k,1}^{y},\cdots,\mathbf{l}_{k,N_l}^{y},\mathbf{l}_{k,1}^{x},\cdots,\mathbf{l}_{k,N_l}^{x},f_j^{\phi}(\mathbf{g}_k^{\phi}),f_j^{\theta}(\mathbf{g}_k^{\theta})] \tag{5.7}$$

Then, a matrix $\mathbf{M}$ is build by stacking all $\mathbf{M}_j$ matrices, from which the corrected CLGM model taking into account real data is derived[1].

5.6.4 Implementation Detail

Auxiliary Database. To the best of our knowledge, the only public database annotating both eye landmark positions and gaze is MPIIGaze [Zhang et al., 2016]. However, it only labels three eye landmarks per image on a subset of the dataset, which is not enough for our framework. Instead, we use the synthetic samples from UnityEyes as an auxiliary database. Concretely, we sample m real eye images from the main database and another m synthetic eye images from the auxiliary database in every training batch. After the feedforward pass, the landmark loss in Eq. 5.6 is only computed on synthetic samples (which have landmark annotations), whereas the gaze loss is only computed on real eye samples, as illustrated in Fig. 5.4. Note that we do not consider the gaze loss on synthetic samples (although they do have gaze ground truth) to avoid a further potential bias towards the synthetic data.

Eye Image Cropping The original UnityEyes samples cover a wide region around the eyes and we need a tighter cropping. To improve the generalization of the network, we give random cropping centers and sizes while cropping UnityEyes samples. Cropped images are then resized to fixed dimensions.

Network Configuration. We set the size of the input images as 36×60. The network is com-

[1]Note that the corrected model relies on real data. In all experiments, the subject(s) used in the test set are never used for computing a corrected CLGM model.

posed of 4 convolutional layers and 6 fully connected layers. The 4 convolutional layers are shared among the predictions of the CLGM coefficients, scale and translation. After the 4 convolutional layers, the network is split into 3 task-specific branches and each branch consists of 2 fully connected layers. Note that the head pose information is also concatenated with the feature maps before the first fully connected layer in the CLGM coefficient branch since the eye shape is also affected by the head pose. The network is learned from scratch in this work.

5.7 Experiment Protocol

5.7.1 Dataset

Two public datasets of real images are used: UTMultiview [Sugano et al., 2014] and Eyediap [Funes Mora et al., 2014].

UTMultiview Dataset. It contains a large amount of eye appearances under different view points for 50 subjects thanks to a 3D reconstruction approach. This dataset provides the ground truth of gaze and head pose, both with large variability. In our experiment, we follow the same protocol as [Sugano et al., 2014] which relies on a 3-fold cross validation.

Eyediap Dataset. It was collected in office conditions. It contains 94 videos from 16 participants. The recording sessions include continuous screen gaze target (CS, small gaze range) and 3D floating gaze target (FT, large gaze range), both based either on a close to static head pose (SP) and mobile head pose (MP) scenario. In experiment, we follow the same person-independent (PI) protocol as [Funes-Mora and Odobez, 2016]. Concretely, for the CS case, we first train a deep network with all the SP-CS subjects but *leave one person out*. Then the network is tested on the left one in both SP-CS and MP-CS sessions (for cross session validation). We do the same for FT case (SP-FT and MP-FT sessions). Note that all eye images are rectified so that their associated head poses are frontal [Funes-Mora and Odobez, 2016].

5.7.2 Synthetic Dataset and CLGM Training

As mentioned above, we use UnityEyes as the auxiliary dataset.

CLGM. For each experimental setting (datasets or sessions), we derive a CLGM model trained from frontal head pose samples using the gaze ranges of this setting. The resulting CLGM models is then further corrected as described in Section 5.6.3.

Auxiliary training samples. For multitask training, the auxiliary synthetic samples are generated with corresponding gaze and head pose ranges matching those of the dataset and session settings.

Synthetic sample refinement. One challenge when training from multiple datasets is the difference in data distribution. Although SimGAN [Shrivastava et al., 2017] has been proposed

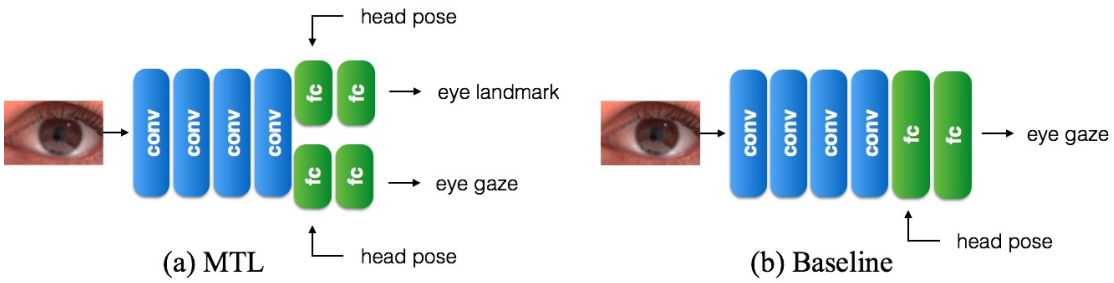


Figure 5.5: Contrast models. (a) MTL architecture. (b) Baseline architecture.

to narrow down the distribution gap between synthetic images and real images, optimizing GAN models is difficult. Without suitable hyper parameters and tricks, the semantic of images after refining can be distorted. In our experiment, we simply adapt the UnityEyes synthetic images to UTMultiview samples through grayscale histogram equalization, and to Eyediap samples by Gaussian blurring.

5.7.3 Model Setup

In terms of gaze estimation models, we considered the models below. The architectures are given in Fig. 5.2 (proposed approach) and in Fig. 5.5 (contrastive approaches). Note that the architectures of the first three models below are the same whenever possible and all the models below are pretrained with synthetic data so that a fair comparison can be made.

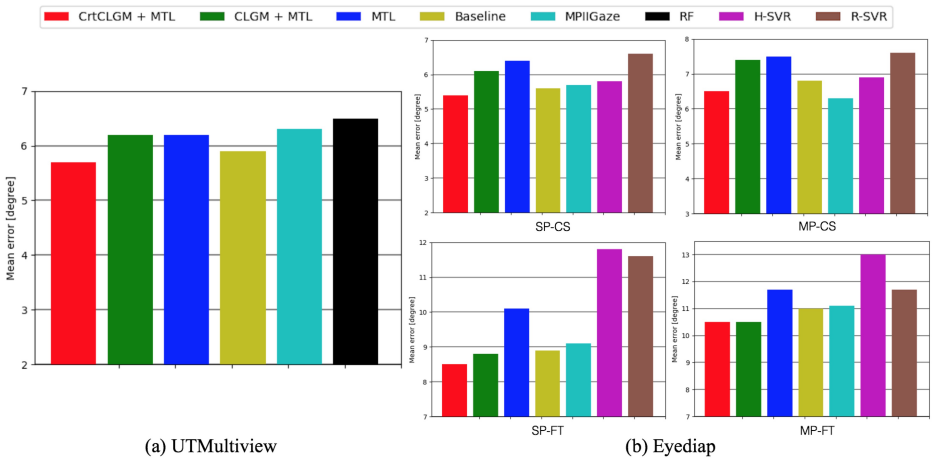
CrtCLGM + MTL. This is our proposed multitask framework based on the corrected CLGM model, see Fig. 5.2.

CLGM + MTL. This model is the same as above (CrtCLGM + MTL), except that the CLGM model is not corrected.

MTL. To contrast the proposed model, we implement a multitask learning network which also predicts landmarks and gaze jointly. Unlike the CrtCLGM + MTL, this model predicts the two labels directly by splitting the network into 2 separate branches after several shared layers. We also forward the head pose information to the features in both branches since both landmarks and gaze are affected by head pose.

Baseline. The baseline model performs a direct gaze regression from the eye appearances using the same base network architecture as in the two previous cases. The head pose is also used in this architecture.

MPIIGaze. For experiments on the Eyediap dataset, we also implemented the network of [Zhang et al., 2016] to allow the comparison of different architectures. For the UTMultiview dataset, we directly report the result of this architecture from [Zhang et al., 2016].



(a) UTMultiview (b) Eyediap

Figure 5.6: Gaze estimation accuracy on UTMultiview and Eyediap.

5.7.4 Performance Measurement

Gaze Estimation. We used the same accuracy measurement as [Funes-Mora and Odobez, 2016] for gaze estimation. The gaze estimation error is defined as the angle between the predicted gaze direction vector and the ground truth gaze direction vector.

Landmark Detection. We also measure the auxiliary task of our model. The GI4E database [Villanueva et al., 2013] is used to test the performance of iris center localization. To apply our method, we first detected facial landmarks with Dlib, then extracted the eye patches using the landmark positions of eye corners (placing the eye in the middle of the cropped image, making the eye width half of the image width, setting the aspect ratio of image as 0.6). Note that the eye images are processed with grayscale histogram equalization. The eye images are then forwarded to the UTMultiview trained network (frontal head pose assumed). In the evaluation, we adopt the maximum normalized error [Gou et al., 2017].

5.8 Results

We report the gaze estimation accuracy of UTMultiview and Eyediap in Fig. 5.6a and Fig. 5.6b respectively. Some qualitative results are demonstrated in Fig. 5.7. Please note that we target at single eye gaze estimation. We think it is not suitable to compare with full face based methods since some datasets (UTMultiview) do not provide the full face and the gaze definition can be different (e.g. gaze fixation points [Krafka et al., 2016] and middle point of face as the origin of gaze direction [Zhang et al., 2016]).

5.8.1 UTMultiview Dataset

From Fig. 5.6a, we note that the proposed CrtCLGM + MTL model shows the best performance (5.7°) among the contrast methods including two state-of-the-art works, MPIIGaze net [Zhang et al., 2016] (6.3°with our implementation) and RF [Sugano et al., 2014] (6.5°). We also note

from Fig. 5.7 that accurate gaze estimation and landmark localization are achieved by our method regardless of eye scale, eye translation and large head pose.

In contrast, we find that the CLGM + MTL model performs worse than the CrtCLGM + MTL model. This is understandable since the optimization of the multitask loss can be difficult if the landmark-gaze correlations are different between the synthetic data and real data. Sometimes the optimization process competes between the two tasks and the final network can be bad for both tasks. This is also shown in Tab. 5.1 where the iris center localization of the CLGM + MTL model is not so accurate. This result demonstrates the importance of the CLGM correction.

When looking at the result of MTL model, it is a bit surprising that its error is on a par with the Baseline method and MPIIGaze net which only target gaze optimization. It thus seems that the MTL model failed to improve gaze estimation through a direct and pure feature sharing strategy. As shown in Fig 5.5a, the landmark positions are regressed from the shared features directly in the landmark branch, which means some information such as eye scale and eye translation are contained in the shared features. Although this geometric information is important to landmark localization, they are irrelevant elements for gaze estimation and might even degrade it. Therefore, our mechanism which decouples the eye scale and translation from eye shape variation is thus necessary and important.

Owing to the reasonable geometric modelling of the scale, translation and head pose, our method also demonstrates superior performance to the Baseline model and the MPIIGaze network. Note that our Baseline model is slightly better than the MPIIGaze net, possibly because in the Baseline, the head pose information is added earlier and thus processed by more layers. Thanks to the large data amount (including the synthetic data used for pretraining), all the network models perform better than the RF (random forest) method.

5.8.2 Eyediap Dataset

Two existing methods H-SVR and R-SVR [Funes-Mora and Odobez, 2016] are used for comparison. From Fig. 5.6b, we note that the proposed CrtCLGM + MTL model achieves the best result in all sessions (5.4°, 6.5°, 8.5°, 10.5°respectively) except the MP-CS session (MPIIGaze: 6.3°). In particular, compared with other methods, the performance improvement is much larger in the floating target (FT) session than in the continuous screen (CS) session, indicating that our method can perform even better for applications requiring gaze in the 3D space, when large head pose and gaze angles are present.

When comparing the results of the CrtCLGM + MTL model and CLGM + MTL model, we note that the former is better for all the sessions which further corroborate the importance of CLGM correction. Compared with the UTMultiview dataset, the MTL model obtains much worse results than other network based methods (especially the Baseline and the MPIIGaze) in Eyediap dataset. Given that the Eyediap samples are much more blurry than the UTMultiview dataset, the direct regression of landmark positions without the constrained model is difficult

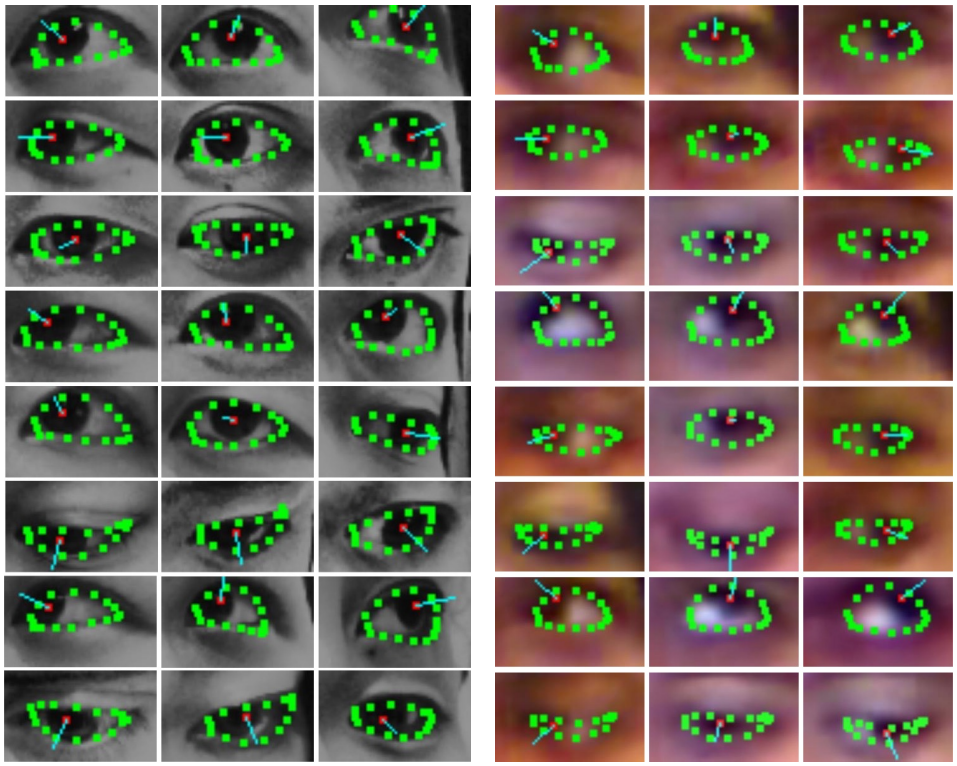


Figure 5.7: Eye landmark detection and gaze estimation results on UTMultiview (***Left***), Eyediap (***Right***). The cyan line represents the estimated gaze direction.

and inaccurate, and the inaccurate landmark detection may confuse the shared architecture and ultimately instead of helping gaze inference, tends to degrade the results. In contrast, our CLGM model is better at handling blurry data thanks to the introduction of an explicit geometrical model, and that learning the parameters of the CLGM model rather than the unconstrained landmark positions provides some form of regularization which prevents the network from overfitting. This also demonstrates the advantage of our method over traditional geometrical approaches where high resolution images are usually required.

When comparing the results across sessions (i.e recording situations, see Sec. 5.7.1), we can observe that the accuracy of the floating target (FT) sessions are worse than the CS screen sessions, which is intrinsically due to the more difficult task (looking at a 3D space target) involving much larger head poses (potentially harming eye image frontalization) and gaze ranges. On the other hand, the results show that our method achieves the most robust performance in cross session validation (train on SP, test on MP).

5.8.3 Iris Center Localization

Lastly, we show the performance of the auxiliary task, landmark detection. Tab. 5.1 reports the accuracy of the iris center localization.

From the table, our method achieves the best performance compared with the state-of-the-art

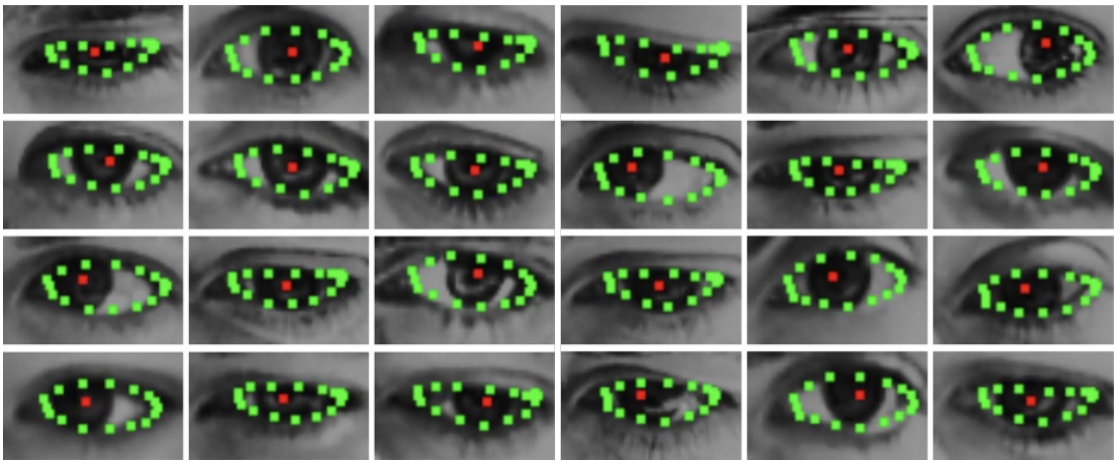


Figure 5.8: Eye landmark detection results on GI4E dataset.

Table 5.1: Iris center localization on GI4E dataset.

Method	$d_{eye} \leq 0.05$(%)	$d_{eye} \leq 0.1$(%)	$d_{eye} \leq 0.25$(%)
Timm and Barth [2011]	92.4	96.0	97.5
Villanueva et al. [2013]	93.9	97.3	98.5
Gou et al. [2017]	94.2	99.1	99.8
CLGM + MTL	92.5	99.7	100
CrtCLGM + MTL	**95.1**	**99.7**	**100**

works in all the three criteria which correspond to the range of pupil diameter, the range of iris diameter and the distance between eye center and eye corner [Timm and Barth, 2011] respectively. Concretely, most of the detections are within the pupil, few of them lie outside the iris and almost all falls inside the eye region. In contrast, the CLGM + MTL model is inferior to the CrtCLGM + MTL one in the $d_{eye} \leq 0.05$(%) measurement, which means more detections of the CLGM + MTL model deviate from the pupil. As discussed in Sec.6.1, it can be explained by the differences in landmark-gaze correlations between the synthetic data and real data.

Some qualitative results are shown in Fig. 5.8. Note that we assumed that the head poses of all the samples were frontal since this label was not provided in this dataset. Even under this assumption we still achieved accurate iris center localization, which demonstrates that our method can be used in a wide scope of eye landmark detection applications where head pose information may not be available.

5.9 Conclusion

In this chapter, we proposed a multitask learning approach for gaze estimation. This approach is based on a Constrained Landmark-Gaze Model which models the joint variation of the eye landmarks and gaze in an explicit way, which helps in (i) solving the absence of annotation on different datasets for some task (in our case, landmarks); (ii) better leveraging in this way the benefits of the multitask approach. This model differs from geometrical methods since

landmarks and gaze are jointly extracted from eye appearance. Experiments demonstrate the capacity of our approach, which is shown to outperform the state-of-the-art in challenging situations where large head poses and low resolution eye appearances are presented. Our idea of CLGM model can also be extended to joint tasks like facial landmark detection and head pose estimation. For instance, using the FaceWarehouse [Cao et al., 2014] dataset as 3D face and landmark statistical model to generate faces with different identities and expressions which can be randomly rotated with different head poses. Since pose and landmarks are correlated, a constrained landmark-head pose model could be built and trained as we propose.

On the hand, although the head pose is not so important for landmark detection as shown in Tab. 5.1, we note from Eq. 5.5 that our model requires precise head pose label for gaze estimation, which may limit the application scope of our method. This problem can be possibly addressed by estimating the head pose from the eye appearance or full face as another task. We leave this as a future work.

6 Few-Shot User-Specific Gaze Adaptation via Gaze Redirection

In this chapter, we address the problem of person-specific gaze model adaptation from only a few reference training samples. The main idea is to use gaze redirection technique to generate more person-specific samples. To achieve this goal, we rely on synthetic data to train a gaze redirection model then adapt it to real data in a self-supervised way. To our best knowledge, it is the first work which applies gaze redirection to gaze estimation.

The rest of this chapter is organized as follows: we motivate our main idea and list the contributions in Chapter. 6.1; Some background on gaze redirection is give in Chapter. 6.2; Chapter. 6.3-6.5 presents the method overview, the gaze redirection approach and person-specific gaze adaptation respectively; The experiment protocol and results are demonstrated in Chapter 6.6 and 6.7 respectively; After a brief discussion in Chapter. 6.8, we summarize our approach in Chapter. 6.9.

6.1 Motivation and Contributions

As mentioned in Chapter. 2, in spite of recent progresses partly due to the use of deep neural networks [Zhang et al., 2016, 2017; Krafka et al., 2016; Fischer et al., 2018; Liu et al., 2018; Park et al., 2018], vision based gaze estimation is still a challenging and open problem because of challenges like lack of data, person-specific bias and systematic bias.

In this work, we focus on the problem of person-specific gaze adaptation which has not received enough attention compared to cross person gaze estimation. More specifically, the aim is to only rely on few samples since collecting training samples for a new subject is expensive. In this context, a first and interesting result that we show is that a direct and simple fine tuning of a neural network gaze regressor can improve person-specific gaze estimation by a good margin, even if the number of person-specific samples is as small as 9. We then propose to further improve the performance of such gaze adaptation method by using as additional training data gaze-redirected samples synthesized from the given reference samples, as illustrated in Fig. 6.1. Compared with domain adaptation methods like SimGAN [Shrivastava

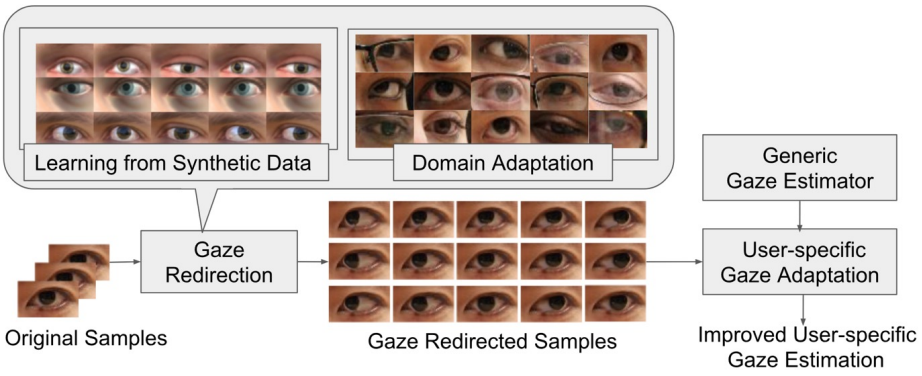


Figure 6.1: Approach overview. A few reference eye images (with gaze ground truth) from a user are used as input to a gaze redirection synthesis module to generate further training samples. The latter (and reference samples) are used to fine-tune a generic gaze estimator to obtain a user-specific gaze estimator.

et al., 2017], which work by retargeting synthetic images into subject specific eye images, we firmly believe that a gaze redirection framework relying on reference eye images and user defined gaze changes (redirection angles) can generate samples with more realistic appearance (since they are directly derived from real eye images of the subject) and more reliable ground truth (less systematic and person-specific bias), thus demonstrating better performance when used for person-specific gaze adaptation. By investigating the above ideas, we make the following contributions:

- **Gaze redirection network training.** Unlike previous approaches [Kononenko et al., 2017; Ganin et al., 2016], our redirection network is pre-trained with synthetic eye images so that a large amount of well aligned image pairs (the same eye position, eye size, head pose and illumination) can be exploited. As a result, thanks to the large amount of data, the network does not require the eye landmarks as anchoring points. Besides, we also propose to exploit the segmentation map of synthetic samples for regularization during training.
- **Gaze redirection domain adaptation.** Training with synthetic data results in the domain shift problem. However, as we do not have aligned pairs of real images to do domain adaptation, we proposed instead a self-supervised method relying on a cycle consistency loss and a gaze redirection loss.
- **Person-specific gaze adaptation using gaze-redirected samples.** We hypothesize that these samples will provide more diverse visual content and gaze ground truth compared to the reference samples they originated from, thus improving the person-specific gaze adaptation. To the best of our knowledge, we are the first to propose this idea and a series of experiments to validate its efficacy.

6.2 Background on Gaze Redirection

As far as we know, the computer vision and graphics based gaze redirection for video-conferencing was first studied in [Zitnick et al., 1999], in which two components are included to solve this task. The first is tracking the user's head pose and eye ball motion, and the second consists of manipulating the head orientation and eye gaze. Following this work, Weiner *et.al.* [Weiner and Kiryati, 2002] evaluated and proved the overall feasibility of gaze redirection in face images via eye synthesis and replacement by integrating the vision and graphical algorithm within a demonstration program. But changes in the eyelid configuration were not considered. Then a simple solution that detects eyes and replaces them with eye images in a front gaze direction was proposed in [Wolf et al., 2010; Qin et al., 2015]. Kononenko *et.al.* proposed a pixel-wise replacement method using an *eye flow tree* and could synthesize realistic views with a gaze systematically redirected upwards by 10 to 15 degrees [Kononenko and Lempitsky, 2015]. Then they updated the eye flow tree by a deep warping network trained on pairs of eye images corresponding to eye appearance before and after the redirection [Ganin et al., 2016; Kononenko et al., 2017]. However, these methods require large amount of annotated data for training.

To circumvent this issue, Wood et al. [2018] proposed a model based method that does not need any training samples. It first builds and fits a multi-part eye region model using an analysis-by-synthesis method to simultaneously recover the eye region shape, texture, pose, and gaze for a given image. Then, it manipulates the eyes by warping the eyelids and rendering eyeballs in the output image. It achieves better results especially for large redirection angles.

6.3 Method Overview

Our overall approach for user-specific gaze adaptation is illustrated in Fig. 6.1. It consists in fine-tuning a generic neural network using labeled training samples. However, rather than only using the very few (less than 10) reference samples, we propose to generate additional samples using a gaze redirection model which is shown in Fig. 6.2. It is composed of the redirection network itself and a domain adaptation module. The left part of Fig. 6.2 illustrates the redirection network which takes the eye image, the user defined redirection angle and the head pose as input. It is designed as an encoder-decoder manner where the output of the decoder is an inverse warping field. The gaze-redirected sample is then generated by warping the input eye image with the predicted inverse warping field (via a differentiable sampler). The right part of Fig. 6.2 is the domain adaptation module which is conducted in a self-supervised way through a cycle consistency loss and a gaze redirection loss.

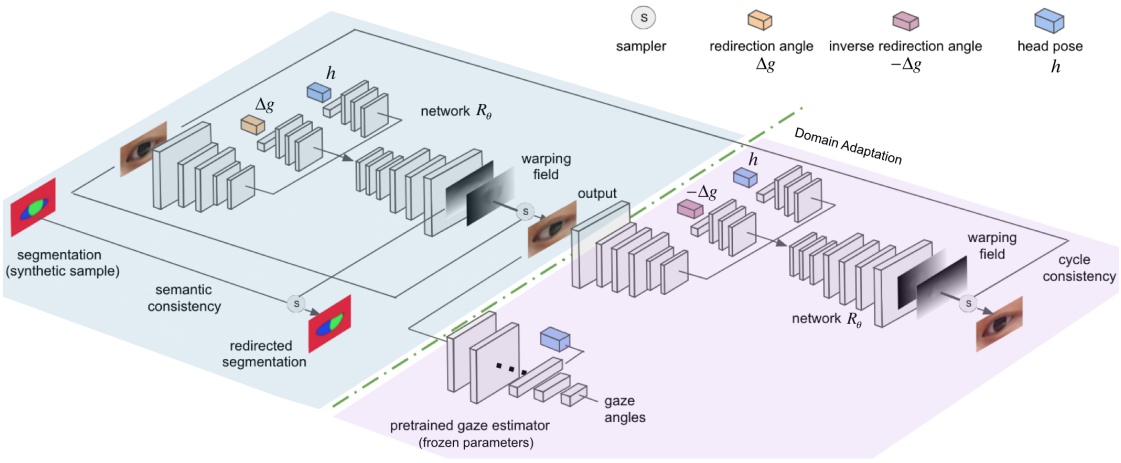


Figure 6.2: Gaze redirection network (top left), along with learning components (eye segmentation for semantic consistency, cycle consistency, gaze prediction consistency).

Figure 6.3: Aligned UnityEyes samples (placed in rows)

6.4 Gaze Redirection

6.4.1 Synthetic Data for Gaze Redirection Learning

In principle, the training of a gaze redirection network needs well aligned image pairs where the two images (the input one and the redirection ground truth for supervision) share the same overall illumination condition, the same person-specific properties (skin color, eye shape, iris color, pupil color) and the same head pose. The only difference should be gaze-related features such as eye ball orientation and eyelid status. This strict requirement make it hard to collect real data. In this paper, we propose to use synthetic samples instead. Concretely, we use the UnityEyes Engine [Wood et al., 2016b] to produce 3K eye image groups, each containing 10 images generated with the same illumination, the same person-specific parameter, the same head pose, but different gaze parameters, as shown in Fig. 6.3. A total of 10*9 image pairs can thus be drawn from each group. In our work, we used 10K image pairs for training.

6.4.2 Gaze Redirection Network

Architecture. It is illustrated in Fig. 6.2. The network takes three variables as input, the eye image $\mathbf{I}$, the head pose $\mathbf{h}$ and the user defined redirection angle $\Delta\mathbf{g}$. Among them, $\mathbf{I}$ is processed by an image branch and encoded as a semantic feature, while $\mathbf{h}$ and $\Delta\mathbf{g}$ are processed with another two branches and encoded as features which will guide the gaze related visual changes. Note that the head pose input is a must since it is one of the elements which determine the appearance of eye images. The three output features are then stacked in a bottleneck layer and further decoded into two inverse warping maps $\mathbf{m}_x$ and $\mathbf{m}_y$:

$$\mathbf{m}_{x,y} = \mathbf{R}_\theta(\mathbf{I}, \Delta\mathbf{g}, \mathbf{h}) \tag{6.1}$$

where $\mathbf{R}$ is the redirection network and θ is the network parameter. Similarly to [Kononenko et al., 2017], we then use a differentiable grid sampler s [Jaderberg et al., 2015] to warp the input image and generate the gaze-redirected image $\mathbf{I}_{\Delta\mathbf{g}}$ whose gaze ground truth is $\mathbf{g}+\Delta\mathbf{g}$ ($\mathbf{g}$ is the gaze of the original image $\mathbf{I}$) according to:

$$\begin{aligned}\mathbf{I}_{\Delta\mathbf{g}}(x,y) = \sum_i \sum_j \mathbf{I}(i,j) \cdot max(0, 1-|i-\mathbf{m}_x(x,y)|) \\ \cdot max(0, 1-|j-\mathbf{m}_y(x,y)|).\end{aligned} \tag{6.2}$$

For simplicity, we rewrite the above formulas as:

$$\mathbf{I}_{\Delta\mathbf{g}} = \mathbf{I} \circ \mathbf{R}_\theta(\mathbf{I}, \Delta\mathbf{g}, \mathbf{h}) \tag{6.3}$$

where $\circ$ represents the warping operation. Compared with direct synthesis, this strategy projects the pixels of the input to the output, which guarantees that the input and the output will share similar color and illumination distributions.

For training, we use an L1 loss to measure the difference between the redirection output $\mathbf{I}_{\Delta\mathbf{g}}$ and the ground truth $\mathbf{G}_\mathbf{I}$. Therefore, generating the required inverse warping field for redirection is learned in an indirect supervised way.

Semantic Consistency. So far, the network can be evaluated by measuring the reconstruction loss between the predicted gaze-redirected eye image $\mathbf{I}$ and the corresponding ground truth $\mathbf{G}_\mathbf{I}$. If the predicted inverse warping field is accurate, then the different semantic parts of the eye (pupil, sclera and background) should also be well redirected. We thus propose to enforce the warping consistency at the semantic level. To do so, for each synthetic image $\mathbf{I}$, we extract the semantic map as follows: we first fit convex shapes to the eyelid landmarks and the iris landmarks (provided by UnityEyes) to get the maps of the iris + pupil region, the sclera region and the background region. We then merge these three maps into a segmentation map $\mathbf{S}_\mathbf{I}$, as shown in Fig. 6.4a. It is important to note that this step is deterministic and is not a part of the network. Then, any segmentation map $\mathbf{S}_\mathbf{I}$ can then be redirected with the inverse warping field $\mathbf{R}_\theta(\mathbf{I}, \Delta\mathbf{g}, \mathbf{h})$ (which is predicted from the original image $\mathbf{I}$) and compared with

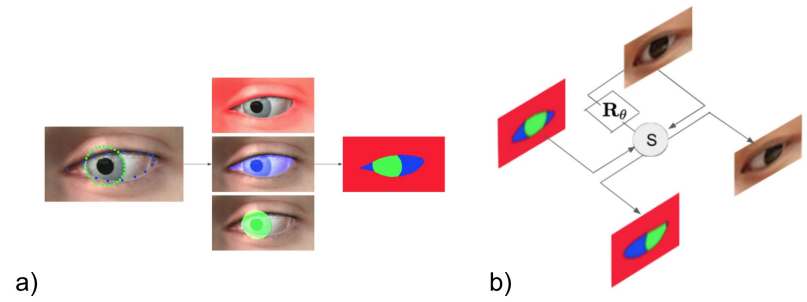


Figure 6.4: Semantic consistency. (a) Deterministic segmentation of a synthetic sample, red: background, blue: sclera, green: iris + pupil. (b) the gaze redirection of a segmentation map.

the segmentation map $\mathbf{S}_{\mathbf{G}_\mathbf{I}}$ of the target redirected eye $\mathbf{G}_\mathbf{I}$.

Overall Loss. According to previous paragraphs, our overall redirection loss L_R (for synthetic data) can be defined as the sum of a reconstruction loss and of the semantic loss, using in each case L1 norms. It is thus defined as:

$$L_R = ||\mathbf{I} \circ \mathbf{R}_\theta(\mathbf{I}, \Delta\mathbf{g}, \mathbf{h}) - \mathbf{G}_\mathbf{I}||_1 + ||\mathbf{S}_\mathbf{I} \circ \mathbf{R}_\theta(\mathbf{I}, \Delta\mathbf{g}, \mathbf{h}) - \mathbf{S}_{\mathbf{G}_\mathbf{I}}||_1 \tag{6.4}$$

Please note that the segmentation map is not processed by the network (looking at Fig. 6.4b) and will not be required at user gaze adaptation time for generating redirected samples.

6.4.3 Domain Adaptation for Gaze Redirection

Because of the domain difference between synthetic and real data, the performance of the network $\mathbf{R}_\theta$ learned only from synthetic data degrades when it is applied to real data. A straightforward solution to solve this issue would be to fine tune $\mathbf{R}_\theta$ with real image pairs. However, as mentioned above, collecting real image pairs for gaze redirection is difficult. In this section, we introduce a self-supervised domain adaptation method relying on two principles. The first one is gaze redirection cycle consistency, and the second one is based on the consistency of the estimated gaze from the gaze redirected image.

Cycle Consistency Loss. It has been used for applications like domain adaptation [Zhu et al., 2017] and identity preserving [Pumarola et al., 2018]. The main idea is that when a sample is transferred to a new domain and then converted back to the original domain, the cycle output should be the same as the input. Similarly, in our case, if a gaze redirected sample $\mathbf{I}_{\Delta\mathbf{g}}$ is further redirected with the inverse redirection angle -$\Delta\mathbf{g}$, the cycle output should be close to the original image $\mathbf{I}$.

In this paper, we apply this cycle consistency scheme to the set of real images, and define the cycle loss as:

$$L_{cycle} = ||\mathbf{I}_{\Delta\mathbf{g}} \circ \mathbf{R}_\theta(\mathbf{I}_{\Delta\mathbf{g}}, -\Delta\mathbf{g}, \mathbf{h}) - \mathbf{I}||_1 \tag{6.5}$$

where $\mathbf{I}_{\Delta\mathbf{g}} = \mathbf{I} \circ \mathbf{R}_\theta(\mathbf{I}, \Delta\mathbf{g}, \mathbf{h})$.

Gaze Redirection Loss. As a weakness, the cycle loss alone could push the redirection network to collapse to an identity mapping (the output of the redirection network is always equal to the input). To prevent this collapse, we propose to exploit a gaze redirection loss. More concretely, given a set of real data, we first train a generic gaze estimator $\mathbf{E}_\phi$ using them. We then freeze the parameters of $\mathbf{E}_\phi$ and use it to define a loss on the gaze-redirected image, enforcing that the gaze predicted from this image should be close to its target ground truth (see bottom of Fig. 6.2). More formally:

$$L_{gaze} = ||\mathbf{E}_\phi(\mathbf{I} \circ \mathbf{R}_\theta(\mathbf{I}, \Delta\mathbf{g}, \mathbf{h})) - (\mathbf{g} + \Delta\mathbf{g})||_2 \tag{6.6}$$

Besides preventing the collapse, the real data trained gaze estimator $\mathbf{E}_\phi$ can help reducing the systematic bias in the gaze redirection network (arising from initially training the network with only synthetic data) and therefore help the domain adaptation of $\mathbf{R}_\theta$.

Network Optimization for Domain Adaptation. To conduct domain adaptation, we do not consider the two losses in the same minibatches, as they are of different nature. In addition, to balance the domain adaptation and the gaze redirection, not all parts of the network need to be adapted simultaneously. In practice, we thus optimize the two losses alternatively according to the following scheme. For the cycle loss L_{cycle}, we only optimize the image encoding branch since i) domain shift usually occurs when encoding an input image into semantic features; ii) the fixed decoder part can further prevent the redirection network from collapsing. For the gaze redirection loss L_{gaze}, only the head pose and gaze branches are updated. The image encoder and decoder remain frozen in this case to prevent an overfitting to L_{gaze}. We use Stochastic Gradient Descent (SGD) to optimize the network.

6.5 Person-Specific Gaze Adaptation

As stated earlier, the aim of the gaze redirection is to generate more person-specific samples for gaze adaptation. In our work, we first train a generic gaze estimator using the real data from several identities. We then adapt the estimator with the samples of a new person and their gaze-redirected outputs. This adaptation is conducted in a few-shot setting, meaning the number of original samples of this new person is few (less than 10). More concretely, the generic estimation network is fine tuned with the person-specific samples during 10 epochs. In the first 5 ones, we use both the original and the gaze-redirected samples, while in the last 5 ones we only use the original samples to minimize the effects of potentially wrong redirected samples. Since the number of samples is small, we use Batch Gradient Descent instead of Stochastic Gradient Descent. Further details about the generic gaze estimator and its adaptation can be found in the Experiment Section.

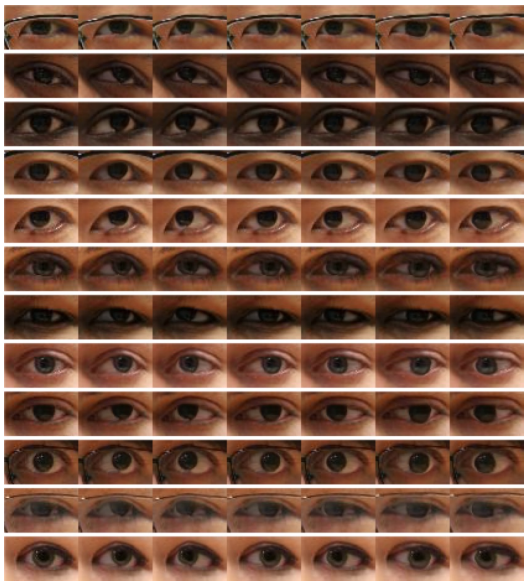
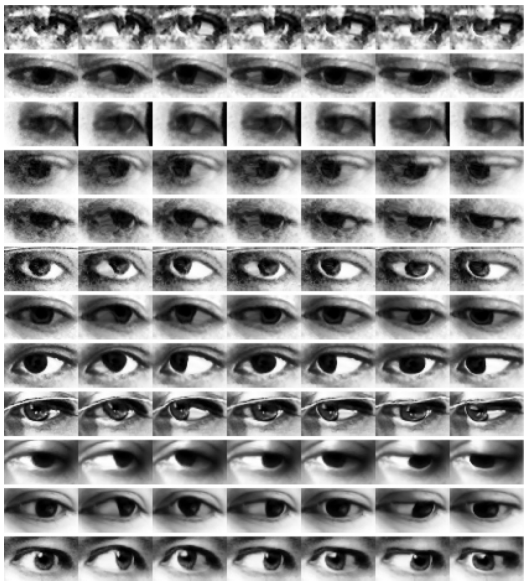
a)  b) 

Figure 6.5: Redirection qualitative results from the ColumbiaGaze (a) and MPIIGaze (b) datasets. In (a) and (b), the first image of each row is an original sample, whereas the remaining images in the row are redirected samples from this original sample.

6.6 Experiment Protocol

In experiments, our main aim is to evaluate the performance of the person-specific gaze estimators adapted from a generic estimator using few reference samples and their gaze-redirected samples. Nevertheless, we also conduct a subjective test to evaluate to which extent the redirected samples are realistic enough for humans. Note that in this paper, we only target single eye image gaze estimation (or redirection), leaving the full-face case as future work.

6.6.1 Dataset

We use the ColumbiaGaze Dataset [Smith et al., 2013] and the MPIIGaze Dataset [Zhang et al., 2015] for experiment. The former one contains the gaze samples of 56 persons while the latter contains eye images of 15 persons.

6.6.2 Generic Gaze Estimator

As our gaze estimator, we use GazeNet [Zhang et al., 2017]. It is is based on a *vgg16* architecture. To train it, we follow the protocols of the ColumbiaGaze and MPIIGaze datasets (i.e. as for cross-subject experiments), using respectively a 5-fold and 15-fold training scheme. The error of our generic gaze estimator on ColumbiaGaze is 3.54° (3.9° in [Park et al., 2018]) while the error on MPIIGaze is 5.35° (5.5° in [Zhang et al., 2017]), showing better performance than the

state-of-the-art results. Please note that the generic gaze estimator[1] is also exploited as $\mathbf{E}_\phi$ to define the gaze redirection loss, as defined in section 6.4.3.

6.6.3 Model Setups

Starting from the generic gaze estimator, we develop a series of adaptation methods to contrast with our approach. The first two methods are the linear (*LinAdap*, [Liu et al., 2018]) and the SVR (*SVRAdap*, using the features of the second last layer [Liu et al., 2018; Krafka et al., 2016]) gaze adaptation methods which learn additional regressors from the gaze estimator output (*LinAdap*) or features (*SVRAdap*), and thus do not change (or adapt) the generic gaze estimator. In contrast, the third and fourth approaches directly fine tune the generic estimator using either only the reference samples (*FTAdap*, *FT* for fine tuning) or as well the gaze redirected samples (*RedFTAdap*, *Red* for redirection).

In addition, we also implement a differential gaze estimator *DiffNet* [Liu et al., 2018] for comparison. The *DiffNet* is trained to predict gaze differences, and it exploits the reference samples to predict the gaze of a new eye image. For a fair comparison, we replace the three convolution layers used as feature extractor in [Liu et al., 2018] with the *vgg16* feature extractor. Please note that the *DiffNet* approach can be regarded as a person-specific network since person-specific samples (at least one) are required to estimating the gaze of new eye image.

6.6.4 Gaze Redirection Parameters

For each person, we randomly draw n (n =1, 5 or 9) person-specific samples and generate $t \cdot n$ gaze-redirected samples where the default value of t is 10. For the MPIIGaze dataset in which the gaze ground truth is continuous, the yaw and pitch components $(\Delta\mathbf{g}_p, \Delta\mathbf{g}_y)$ of the redirection angle $\Delta\mathbf{g}$ are randomly chosen with the range $[-10,10] \times [-15,15]$ ($[-10,10]$ for pitch, and $[-15,15]$ for yaw). For the ColumbiaGaze dataset, where the annotated gaze is discrete, $\Delta\mathbf{g}$ is chosen from the same range but with discrete values $(\pm 5^\circ, \pm 10^\circ, \pm 15^\circ)$. The impact of t and of the redirection ranges are further studied in the result section.

6.6.5 Performance Measurement.

We use the angle (in degree) between the predicted gaze vector and the ground truth gaze vector as the error measurement. Note that gaze vectors are 3D unit vectors constructed from the pitch and yaw angles. To eliminate random factors, we performed 10 rounds of person-specific sample selection, gaze redirection and gaze adaptation, and reported the average estimation error.

[1] A generic estimator is trained for each fold. In none of the experiments, data from the test subject is used in either part of the training phase.

Table 6.1: ColumbiaGaze dataset: gaze adaptation performance (error in degree)

error / approach / #sample	*Cross Subject*	*LinAdap*	*SVRAdap*	*FTAdap*	*DiffNet*	*RedFTAdap*
1		-	-	5.53	4.64	**3.92**
5	3.54	4.65	7.67	3.11	3.63	**2.88**
9		3.78	5.39	2.79	3.50	**2.60**

Table 6.2: MPIIGaze dataset: gaze adaptation performance (error in degree)

error / approach / #sample	*Cross Subject*	*LinAdap*	*SVRAdap*	*FTAdap*	*DiffNet*	*RedFTAdap*
1		-	-	5.28	5.93	**4.97**
5	5.35	5.43	7.68	4.64	4.42	**4.20**
9		4.61	5.79	4.31	4.20	**4.01**

6.7 Results

6.7.1 Qualitative Results of Gaze Redirection

We show some qualitative results of the redirection network in Fig. 6.5(a) and (b). As can be seen, our redirection network does a realistic synthesis for samples with different skin or iris color. Furthermore, we also found that the redirection model is robust when working with noisy eye images, as illustrated in several rows of Fig. 6.5(b).

6.7.2 Performance of Gaze Adaptation

The results of person-specific gaze estimation are reported in Tab. 6.1 (ColumbiaGaze dataset) and Tab. 6.2 (MPIIGaze dataset). From the tables, we observe that the proposed approach *RedFTAdap* achieves the best results while the *LinAdap* and *SVRAdap* methods obtain the worst results, sometimes even degrading the generic gaze estimator. The unsatisfactory performance of the latter models (*LinAdap* and *SVRAdap*) is probably due to the fact that the linear and SVR regressor do not make changes to the generic gaze estimator and thus the capacity of gaze adaptation is limited. We also find that the *DiffNet* is not always superior to the simpler *FTAdap* approach. This is surprising and shows that the ability of direct network fine tuning with small amount of data (less than 10) is often overlooked in the literature and not even unattempted. To the best of our knowledge, we are the first to report this result which can inspire new research on user-specific gaze estimation.

When comparing *RedFTAdap* with the best results of *DiffNet* and *FTAdap*, we note that our approach leads the performance by around 0.2°. While this may seem a marginal improvement, a more detailed analysis of the results shows that our approach improves the results of **84.2%** of the subjects from the ColumbiaGaze dataset and of **80%** of the subjects from the MPIIGaze dataset (compared with the best results of *both DiffNet* and *FTAdap*), which means that the improvements brought by *RedFTAdap* are stable and rather systematic.

Table 6.3: ColumbiaGaze: Results with different redirection range (error in degree)

error $\Delta\mathbf{g}_p$ \ $\Delta\mathbf{g}_y$	$[-5,5]$	$[-10,10]$	$[-15,15]$
$[-10,10]$	2.66	2.62	**2.60**

Table 6.4: MPIIGaze: Results with different redirection range (error in degree)

error $\Delta\mathbf{g}_p$ \ $\Delta\mathbf{g}_y$	$[-5,5]$	$[-10,10]$	$[-15,15]$
$[-5,5]$	4.15	4.06	4.02
$[-10,10]$	4.10	4.03	**4.01**

From the two tables, we note that the performances of all the methods improve as the number of reference samples increases. We can also notice that our approach seems to have a larger advantage when the number of reference samples is small, demonstrating that the diversity introduced by our redirected samples is more important when fewer person-specific gaze information is provided.

Finally, while in general adaptation methods improve results, we observe on the ColumbiaGaze dataset that they all perform worse than the generic estimator (cross-subject result) when using only one reference sample. This is most probably due to the large variance of the head pose in this dataset, which makes it difficult to learn (through adaptation) person-specific characteristics from only one sample.

6.7.3 Impact of Redirection Range

We use different gaze redirection ranges to generate samples for gaze adaptation. The selected redirection ranges are shown in Tab. 6.3 and Tab. 6.4. Note that we only use one redirection range of pitch for the ColumbiaGaze dataset since the gaze ground truth in this dataset is discrete and there are only three values for the pitch angle, $-10^\circ, 0^\circ, 10^\circ$. It is thus not necessary to produce samples with new ground truth. From the results, we find that larger redirection ranges do bring an improvement, especially for the MPIIGaze dataset where the performance improves from 4.15° to 4.01°. This result is expected since a larger redirection range will usually bring more gaze diversity, provided that the redirection module produces synthesized samples realistic enough for the given user. Besides, we also find from Tab. 6.4 that a larger redirection range for the yaw angle seems to be more effective than a larger redirection range for the pitch.

6.7.4 Impact of Number t

To study the impact of this parameter (the default value was 10 in all other experiments), we randomly selected 9 reference samples for each person and generated $9 \cdot t$ gaze redirected samples, varying t between 0 and 100. We then adapted the generic gaze estimator with these samples as in all other experiments. The corresponding performances are plotted in Fig. 6.6

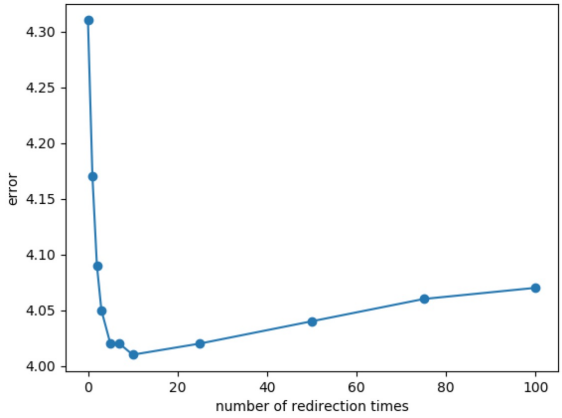


Figure 6.6: Gaze adaptation performances w.r.t redirection times t (MPIIGaze dataset, error in degree).

Table 6.5: Impact of the gaze redirection network domain adaptation (ColumbiaGaze dataset, error in degree).

error / approach / #sample	*FTAdap*	*RedFTAdap-noDA*	*RedFTAdap*
1	5.53	4.35	**3.92**
5	3.11	3.01	**2.88**
9	2.79	2.73	**2.60**

for the MPIIGaze dataset (note that we do not use the ColumbiaGaze dataset since its ground truth and redirection angles are discrete, which limits the number of generated data).

The curve in Fig. 6.6 starts from $t = 0$ (which means only the initial reference samples are used for adaptation). As can be seen, the error decreases rapidly at first when $t \in [0,5]$, remains at a relatively stable point within the range $t \in [5,25]$, and then progressively degrades beyond that. This curve shows that when $t \simeq 10$, the generated samples provide enough diversity to adapt the network, whereas beyond that, the use of too many samples results in an overfit of the network to the generated data which might not reflect the actual distribution of eye gaze appearance of the user.

Table 6.6: Impact of the gaze redirection network domain adaptation (MPIIGaze dataset, error in degree).

error / approach / #sample	*FTAdap*	*RedFTAdap-noDA*	*RedFTAdap*
1	5.28	4.99	**4.97**
5	4.64	4.22	**4.20**
9	4.31	4.04	**4.01**

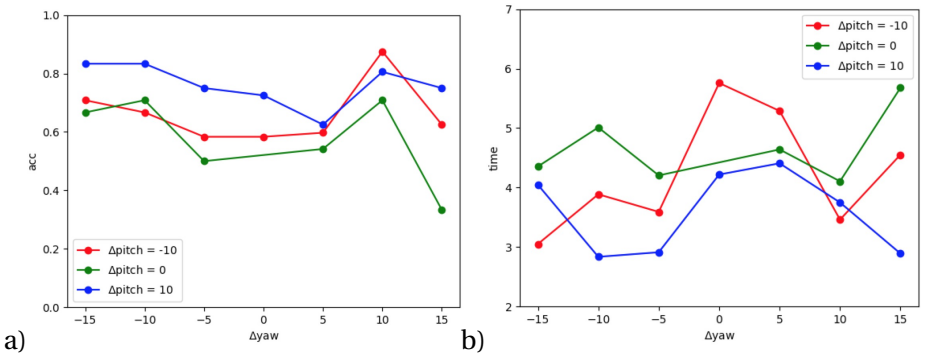


Figure 6.7: Subjective test. (a) decision accuracy w.r.t redirection angles. (b) decision time w.r.t redirection angles.

Figure 6.8: Sample pairs for subjective test. In each pair, the left image is an original image from the dataset, while the right one is a redirected sample (obtained from another original sample) which has the same gaze label (i.e. direction) as the left one.

6.7.5 Impact of Domain Adaptation.

We remove the whole **D**omain **A**daptation step from the redirection network and report the corresponding gaze adaptation results (*RedFTAdap-noDA*) in Tab. 6.5 and Tab. 6.6. On one hand, surprisingly, we note that exploiting the redirection network learned only from synthetic data still helps improving the gaze adaptation process (*FTAdap* vs *RedFTAdap-noDA*). On the other hand, when comparing *RedFTAdap-noDA* and *RedFTAdap*, we find that the domain adaptation further improves the gaze adaptation results. This is particularly the case for the ColumbiaGaze dataset. A possible reason why the domain adaptation is less useful on the MPIIGaze dataset is that the domain difference between MPIIGaze and the synthetic data (all processed with histogram equalization to match MPIIGaze) is comparatively smaller.

6.7.6 Subjective Test

To evaluate whether the gaze redirected samples are realistic, we invited 24 participants for a subjective test. During the test, participants were shown 50 pairs of ColumbiaGaze samples, where one image of the pair did correspond to an actual real data sample, and the second one was a gaze redirected sample. Note that as a result, the eyes in each image pair share the same identity, the same gaze and the same head pose. Some pairs are illustrated in Fig. 6.8 where the real images are all placed on the left for the purpose of demonstration. In the test, the

places of the real and redirected images were selected at random. Participants were asked to choose the sample which they think was real. A software was recording their choices as well as the time they took to make the decisions.

Results are as follows. The average accuracy of making a correct choice is 66%, showing that distinguishing genuine samples from redirected ones is difficult. This is further confirmed by the average time to reach a decision, which is around 4 seconds and shows that people have to take some time to make a careful decision.

We also plot the decision precision and the decision time w.r.t redirection angles in Fig. 6.7. From Fig. 6.7a, we find a general and expected trend that comparing samples with smaller redirection angles leads to more confusion, i.e. a low accuracy (and although as an artefact, the accuracy declines when $\Delta yaw = 15$). The same trend is observed in Fig. 6.7b, where a smaller redirection angle corresponds to a longer decision time. Nevertheless, in general, more participants and samples should be used to confirm these results, which we leave as a future work.

6.8 Discussion

In this section, we discuss techniques we attempted when developing the approach.

More realistic redirected samples. Ganin et al. [2016] used a lightness correction refinement module on the gaze image redirected from the inverse warping field to produce a more realistic final redirected image. It indeed removed a lot of artifacts in our case. However, we found out that it was also degrading the performance of gaze adaptation, because the refinement through a set of convolutional layers was altering too much the distribution of color and illumination.

GAN. We also attempted to use GAN (or CycleGAN when combined with the cycle loss) for domain adaptation. However, as our redirected images are already of high quality, the GAN did not further improve the gaze adaptation step.

6.9 Conclusion

We proposed to improve the adaptation of a generic gaze estimator to a specific person from few shot samples via gaze redirection synthesis. To do so, we first designed a redirection network that was pretrained from large amounts of well aligned synthetic data, making it possible to predict accurate inverse warping fields. We then proposed a self-supervised method to adapt this model to real data. Finally, for the first time to the best of our knowledge, we exploited the gaze redirected samples to improve the performance of a person-specific gaze estimator. Along this way, as a minor contribution, we also showed that the simple fine tuning of a generic gaze estimation network using a very small amount of person-specific

samples is very effective.

Notwithstanding the obtained improvements, a limitation of our method is that the redirection synthesis is not good enough for large redirection angles. It hinders further improvements of gaze adaptation because generated samples can not cover the full space of gaze directions and illumination conditions. We leave gaze redirection with larger angles and more illumination variabilities as future work.

7 Conclusion

7.1 Contributions

In this thesis, we investigated the problem of head nod detection and gaze estimation, two subtle but important non-verbal behaviours to various areas such as Psychology and Sociology, Human Computer and Human Robot Interaction, Virtual Reality and so on. Based on the previous approaches, we proposed and validated several innovative solutions which addresses some weaknesses of prior works.

In the following, we will recall in detail the thesis contributions. The limitations and perspectives of our proposed would also be discussed.

Head Pose Pose Estimation. Head pose estimation is a fundamental task for both head nod detection and gaze estimation. To overcome the weakness of previous 3DMM based methods (they require frontal or mid-profile poses since the 3DMM only models the face region), we combine the strengths of a 3DMM model fitted online with a prior-free reconstruction of a 3D full head model providing support for pose estimation from any viewpoint. Besides, we also propose a symmetric regularization for accurate 3DMM fitting and a tracking initialization strategy to deal with fast motions. The experiment validates our method in unconstrained free settings. A public head pose dataset is also proposed in this chapter.

Head Nod Detection. Based on the obtained head pose, we propose a nod detection approach by making the following two contributions. First, the head motion dynamics are computed within the head coordinate system instead of the world coordinate system, which results in camera pose invariant motion features. Second, we also propose a feature which relates to the head rotation axis, leading to body motion invariant features. The experiment demonstrates the validity of our approach in conversation scenarios.

Deep Multitask Gaze Estimation. In this chapter, we attempt to apply multitask learning to gaze estimation. Concretely, we relate the explicit visual information (eye landmarks) to the more abstract gaze values by a Constrained Landmark-Gaze Model (CLGM), which makes

the learning of gaze much easier. Besides, we also designed a network inferring the CLGM parameters which are later used by a geometric decoder to recover the eye shape and gaze. Our framework decouples gaze estimation from irrelevant geometric variations and thorough experiments demonstrate that our method can achieve competitive results on both gaze estimation and eye landmark localization.

Few-Shot User-Specific Gaze Adaptation. In this chapter, we target at the problem of person-specific gaze adaptation with only a few reference training samples. The main idea to generate more specific samples with the gaze redirection technique. The contributions we made are threefolds: i) we train the gaze redirection model based on synthetic data which provides large amounts of aligned data; ii) we used a self-supervised approach to adapt the gaze redirection model to real data; iii) we used gaze redirection to generate more person-specific samples to improve the performance of person-specific gaze adaptation. Extensive experiments on two public datasets prove the efficacy of our approach.

7.2 Limitations and Perspectives

We then briefly summarize the limitations of our proposed solutions. Some possible future works would also be discussed.

Head Pose Estimation. Although accurate and robust head pose estimation can be achieved by our approach in many difficult scenarios. Our solution is still challenged by the situation where long hair is moving around. Therefore, to further improve our approach, recovering the semantic segmentation of the head model (eg which region is face or hair) can be an interesting perspective. In addition, the current head pose estimation is only based on the observed head. In fact, some other semantic cues such as shoulder or body orientation can also be used to improve the performance, especially for extreme cases. Finally, our work can also be expanded to other tasks. For example, facial expression estimation and analysis under extreme head poses and variations.

Head Nod Detection. To detect a head nod, our method requires the head dynamics from the past frames and the future frames, which means our method can not be used for online nod detection. Therefore, one possible future direction is to develop a framework which enables online nod detection. Besides, we also expect to extend our method to the detection of other head gestures such as head shake or some more complex gestures.

Deep Multitask Gaze Estimation. One limitation of the multitask gaze estimation approach is that it requires precise head pose label for gaze estimation, which may limit the application scope of our method. This problem can be possibly addressed by estimating the head pose from the eye appearance or full face as another task, which we leave as a future work. Besides, our idea of CLGM model can also be extended to joint tasks like facial landmark detection and

head pose estimation, for instance, by building a constrained landmark-head pose model.

Few-Shot User-Specific Gaze Adaptation. The limitation of this work is that the redirection synthesis is not good enough for large redirection angles, which might hinders the further improvements of gaze adaptation since the diversity of generated samples is limited. We leave gaze redirection with larger angles as a future work.

Lastly, we discuss other challenges of Head Nod Detection and Gaze Estimation facing the research community and propose some future directions.

Head Nod Detection. The head nod is an up-and-down head motion performed around the neck. But only capturing the head motion is not enough to detect the head nod. For instance, other head motions like looking down or looking up are also up-and-down movements around the neck. Therefore, other semantic information is needed to correctly detect the head nod in a social interaction scenario. One solution is to extract the gaze features for head nod detection since people tend to stare at the objects (or the other person) while performing head nod.

Gaze Estimation. Eye patch extraction is an important preprocessing step of gaze estimation. However, accurate and robust eye patch extraction is difficult despite the progress being made by head pose estimation and facial landmark detection (especially for large head poses). But we note that there are few works which model this preprocessing step along with eye gaze estimation (most literatures deal with cropped eyes only) and evaluate how the eye patch extraction could affect the gaze estimation. Estimating facial landmarks and gaze together could be one possible solution.

Besides, the 3D gaze annotation in current datasets can be noisy, due to (i) measurement errors: most datasets compute the 3D line of sight by visually estimating the 3D positions of eyes and gaze targets; (ii) participant distractions or blinks, leading to totally wrong annotations. This problem does not receive enough attention from the gaze community and developing methods which learn gaze from noisy labels can be one future direction.

Appendix

A1. Architecture of CLGM based Multitask Network

The detailed architecture of CLGM based Multitask Network (Chapter 5) is illustrated in Fig. 1. Note that the input image resolution is 36*60.

A2. Architecture of Gaze Redirection Network

The detailed architecture of Gaze Redirection Network (Chapter 6) is illustrated in Fig. 2. Note that the input image resolution is 36*60.

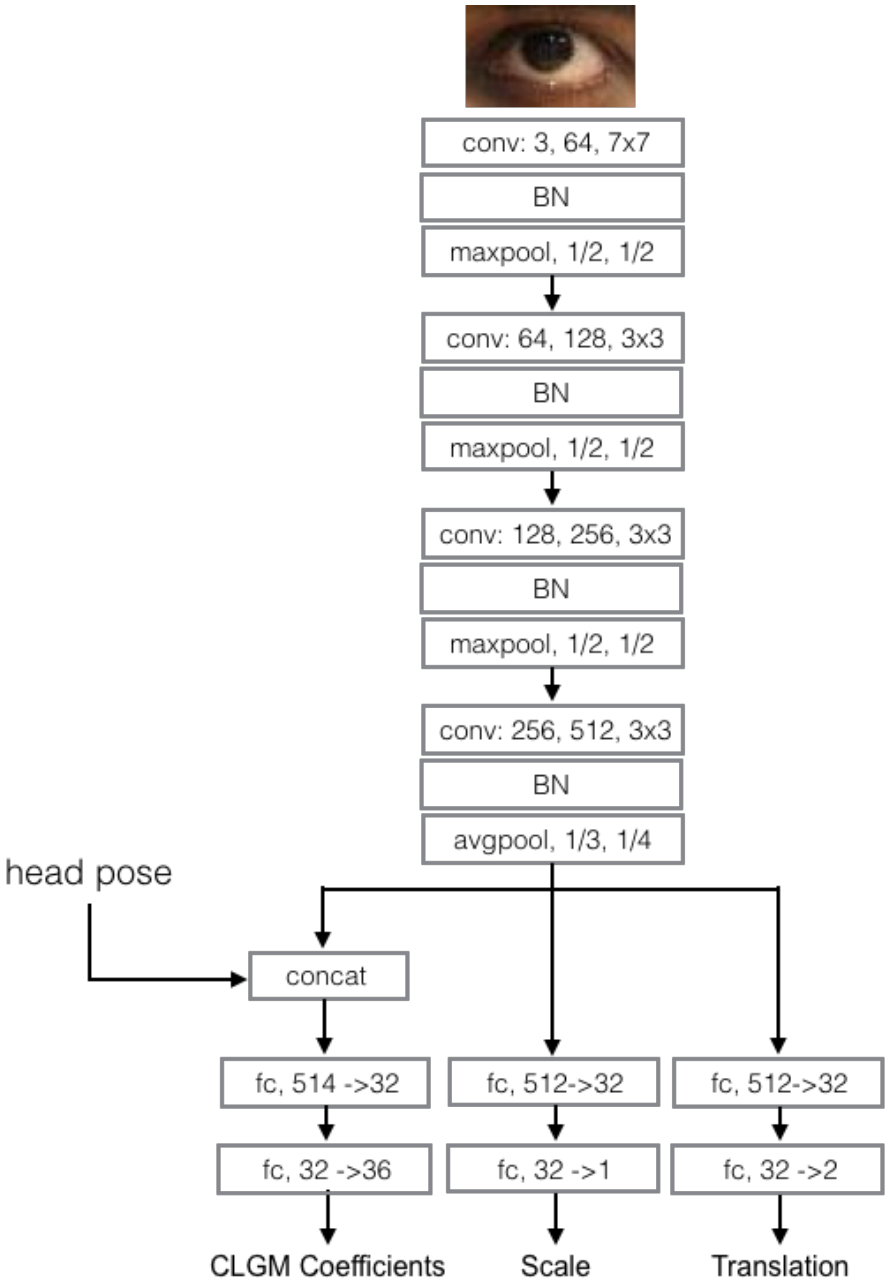


Figure 1: Architecture of CLGM based Multitask Network

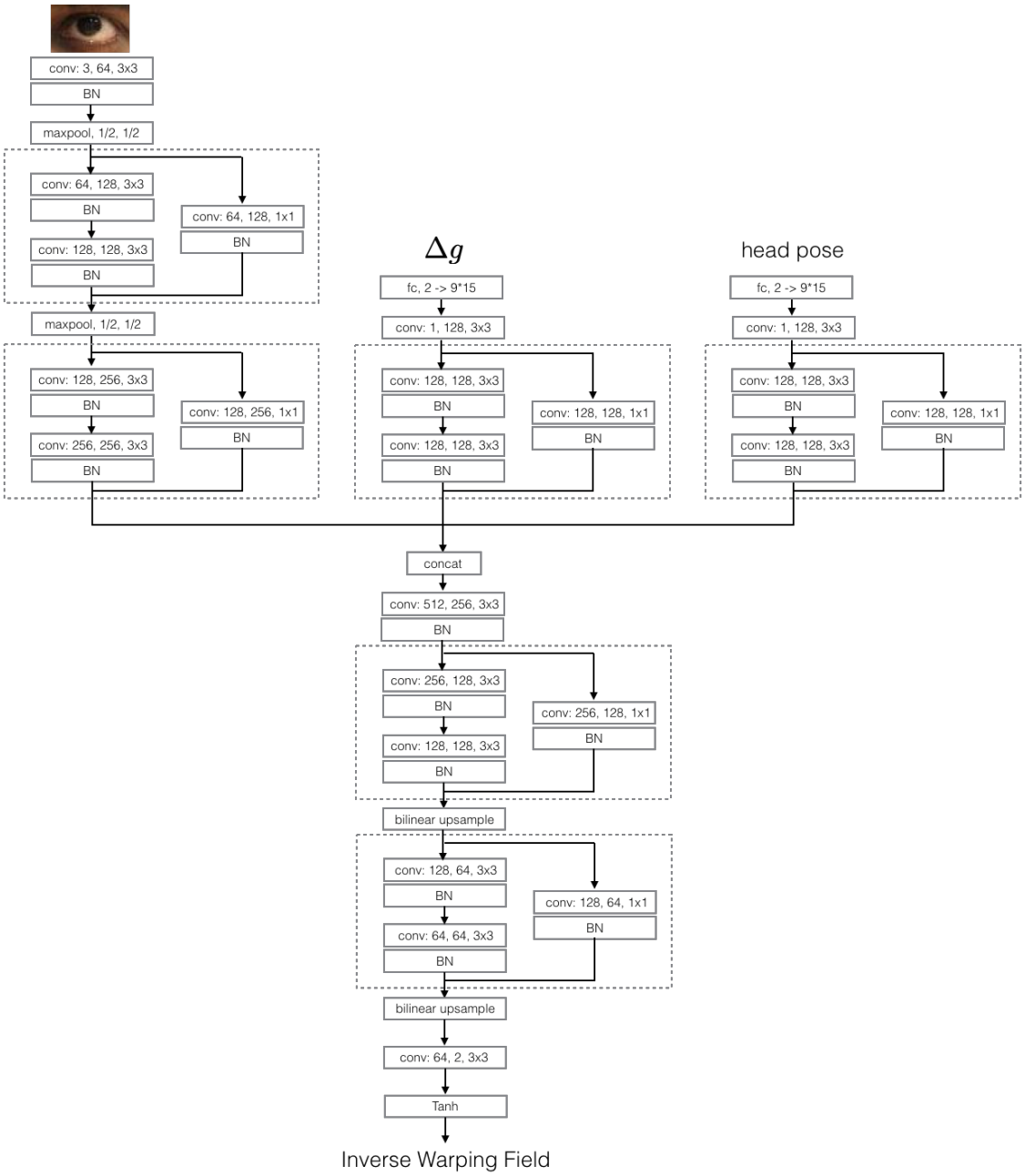


Figure 2: Architecture of Gaze Redirection Network

Bibliography

A.Kapoor and R.Picard (2001). A real-time head nod and shake detector. *In Proceedings from the Workshop on Perspective User Interfaces*, pages 1–5.

Amberg, B., Knothe, R., and Vetter, T. (2008). Expression invariant 3d face recognition with a morphable model. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6.

Andrist, S., Tan, X. Z., Gleicher, M., and Mutlu, B. (2014). Conversational gaze aversion for humanlike robots. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, pages 25–32, New York, NY, USA. ACM.

Ba, S. and Odobez, J.-M. (2008). Multi-party focus of attention recognition in meetings from head pose and multimodal contextual cues. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Las-Vegas.

Baltrusaitis, T., Robinson, P., and Morency, L.-P. (2013). Constrained local neural fields for robust facial landmark detection in the wild. In *ICCV Workshop*.

Baltrušaitis, T., Robinson, P., and Morency, L.-P. (2016). Openface: an open source facial behavior analysis toolkit. In *WACV*.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256.

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *SIGGRAPH*.

Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., and Dunaway, D. (2016). A 3d morphable model learnt from 10,000 faces. In *CVPR*.

Bouaziz, S., Wang, Y., and Pauly, M. (2013). Online modeling for realtime facial animation. *ACM Trans. Graph.*, 32(4).

Cao, C., Weng, Y., Zhou, S., Tong, Y., and Zhou, K. (2014). Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425.

Cao, Y., Canévet, O., and Odobez, J.-M. (2018). Leveraging convolutional pose machines for fast and accurate head pose estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1089–1094. IEEE.

Chen, M., Jin, Y., Goodall, T., Yu, X., and Bovik, A. C. (2019). Study of 3d virtual reality picture quality. In *arXiv*.

Chen, Y. and Medioni, G. (1992). Object modeling by registration of multiple range images. *Image Vision Comput.*, 10:145–155.

Chen, Y., Yu, Y., and Odobez, J.-M. (2015). Head nod detection from a full 3d model. In *Proceedings of the ICCV Workshops*, pages 528–536.

Chen, Z. and Shi, B. E. (2019a). Appearance-based gaze estimation using dilated-convolutions. In *arXiv*.

Chen, Z. and Shi, B. E. (2019b). Appearance-based gaze estimation via gaze decomposition and single gaze point calibration. In *arXiv*.

Cheng, Y., Lu, F., and Zhang, X. (2018). Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *The European Conference on Computer Vision (ECCV)*.

Continente, A., Vondrick, C., Khosla, A., and Torralba, A. (2017). Following gaze in video. In *2017 IEEE International Conference on Computer Vision (ICCV)*.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models: Their training and application. *Comput. Vis. Image Underst.*, 61(1).

Cristinacce, D. and Cootes, T. (2006). Feature detection and tracking with constrained local models. In *Pattern Recognit.*, volume 41, pages 929–938.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *SIGGRAPH*.

D.Cristinacce and T.F.Cootes (2007). Automatic Feature Localisation with Constrained Local Models. *Pattern Recognition*, 41(10):3054–3067.

Derkach, D., Ruiz, A., and Sukno, F. M. (2017). Head pose estimation based on 3-d facial landmarks localization and regression. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 820–827.

Derkach, D., Ruiz, A., and Sukno, F. M. (2019). Tensor decomposition and non-linear manifold modeling for 3d head pose estimation. *International Journal of Computer Vision*, 127(10):1565–1585.

Dias, P. A., Malafronte, D., Medeiros, H., and Odone, F. (2019). Gaze estimation for assisted living environments. In *arXiv*.

Fanelli, G., Dantone, M., Gall, J., Fossati, A., and Van Gool, L. (2013). Random forests for real time 3d face analysis. *Int. J. Comput. Vision*, 101(3):437–458.

Fanelli, G., Gall, J., and Gool, L. V. (2011). Real time head pose estimation with random regression forests. In *CVPR*.

Finnerty, A. N., Muralidhar, S., Nguyen, L. S., Pianesi, F., and Gatica-Perez, D. (2016). Stressful first impressions in job interviews. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, pages 325–332, New York, NY, USA. ACM.

Fischer, T., Chang, H. J., and Demiris, Y. (2018). RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. *European Conference on Computer Vision (ECCV)*.

Funes Mora, K. A., Monay, F., and Odobez, J.-M. (2014). Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*, pages 255–258.

Funes-Mora, K. A. and Odobez, J.-M. (2012). Gaze Estimation from Multimodal Kinect Data. *CVPR Workshop*.

Funes Mora, K. A. and Odobez, J.-M. (2014). Geometric Generative Gaze Estimation (G3E) for Remote RGB-D Cameras. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1773–1780.

Funes-Mora, K. A. and Odobez, J.-M. (2016). Gaze estimation in the 3d space using rgb-d sensors, towards head-pose and user invariance. *International Journal of Computer Vision (IJCV)*, 118(2):194–216.

Ganin, Y., Kononenko, D., Sungatullina, D., and Lempitsky, V. (2016). DeepWarp: Photorealistic image resynthesis for gaze manipulation. *European Conference on Computer Vision (ECCV)*, pages 311–326.

Geng, X. and Xia, Y. (2014). Head pose estimation based on multivariate label distribution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. In *arXiv*.

Gou, C., Wu, Y., Wang, K., Wang, F.-Y., and Ji, Q. (2016). Learning-by-synthesis for accurate eye detection. *ICPR*.

Gou, C., Wu, Y., Wang, K., Wang, K., Wang, F. Y., and Ji, Q. (2017). A joint cascaded framework for simultaneous eye detection and eye state estimation. *Pattern Recognition*, 67:23–31.

Hadar, U., Steiner, T., Grant, E., and Rose, F. (1983). Kinematics of head movements accompanying speech during conversation. *Human Movement Science*, 2:35–46.

Hansen, D. W. and Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(3):478–500.

Hsieh, P.-L., Ma, C., Yu, J., and Li, H. (2015). Unconstrained realtime facial performance capture. In *CVPR*.

Huang, M. X., Li, J., Ngai, G., and Leong, H. V. (2016). Stressclick: Sensing stress from gaze-click patterns. In *Proceedings of the ACM on Multimedia Conference (ACMMM)*, pages 1395–1404.

Huang, Q., Veeraraghavan, A., and Sabharwal, A. (2017). Tabletgaze: unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications (MVAP)*.

Ishikawa, T., Baker, S., Matthews, I., and Kanade, T. (2004). Passive driver gaze tracking with active appearance models. Technical Report CMU-RI-TR-04-08, Carnegie Mellon University, Pittsburgh, PA.

Itoh, T. D., Kubo, T., Ikeda, K., Maruno, Y., Ikutani, Y., Hata, H., Matsumoto, K., and Ikeda, K. (2019). Towards generation of visual attention map for source code. In *arXiv*.

Jaderberg, M., Simonyan, K., Zisserman, A., and kavukcuoglu, k. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2017–2025.

J.Allwood and L.Cerrato. (2003). A study of gestural feedback expressions. *In Proceedings of the First Nordic Symposium on Multimodal Communication*.

Jourabloo, A. and Liu, X. (2016). Large-pose face alignment via cnn-based dense 3d model fitting. In *CVPR*.

Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *CVPR*, pages 1867–1874. IEEE Computer Society.

Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., and Kolb, A. (2013). Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *International Conference on 3D Vision*.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758.

Kononenko, D., Ganin, Y., Sungatullina, D., and Lempitsky, V. S. (2017). Photorealistic Monocular Gaze Redirection Using Machine Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–15.

Kononenko, D. and Lempitsky, V. (2015). Learning to look up: Realtime monocular gaze correction using machine learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4667–4675.

Konrad, R., Angelopoulos, A., and Wetzstein, G. (2019). Gaze-contingent ocular parallax rendering for virtual reality. In *arXiv*.

Krafka, K., Khosla, A., Kellnhofer, P., and Kannan, H. (2016). Eye Tracking for Everyone. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2176–2184.

Li, H., Lin, Z., Shen, X., Brandt, J., and Hua, G. (2015). A convolutional neural network cascade for face detection. In *CVPR*.

Lindén, E., Sjöstrand, J., and Proutiere, A. (2018). Learning to personalize in appearance-based gaze tracking. In *arXiv*.

Liu, G., Yu, Y., Funes-Mora, K. A., and Odobez, J.-M. (2018). A Differential Approach for Gaze Estimation with Calibration. *British Machine Vision Conference (BMVC)*.

Liu, G., Yu, Y., Mora, K. A. F., and Odobez, J. (2019). A differential approach for gaze estimation. *accepted in IEEE Transaction on Pattern Analysis and Machine Intelligence*.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*.

Lu, F., Sugano, Y., Okabe, T., and Sato, Y. (2011). Inferring human gaze from appearance via adaptive linear regression. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 153–160.

Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., and Feris, R. S. (2016). Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. *CoRR*, abs/1611.05377.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*.

Martinez, F., Carbone, A., and Pissaloux, E. (2012). Gaze estimation using local features and non-linear regression. *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1961–1964.

Meyer, G. P., Gupta, S., Frosio, I., Reddy, D., and Kautz, J. (2015). Robust model-based 3d head pose estimation. In *ICCV*.

Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. (2016). Cross-stitch networks for multi-task learning. *CoRR*, abs/1604.03539.

Morency, L., Sidner, C., Lee, C., and Darrell, T. (2007). Head gestures for perceptual interfaces: The role of context in improving recognition. *Artificial Intelligence*, pages 568–585.

Morency, L.-P., Whitehill, J., and Movellan, J. (2008). Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation. In *Face and Gesture*.

Muralidhar, S., Nguyen, L. S., Frauendorfer, D., Odobez, J.-M., Schmid Mast, M., and Gatica-Perez, D. (2016). Training on the job: Behavioral analysis of job interviews in hospitality. In *ICMI*.

Muralidhar, S., Schmid Mast, M., and Gatica-Perez, D. (2017). How may i help you? behavior and impressions in hospitality service encounters. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, pages 312–320, New York, NY, USA. ACM.

Müller, P., Buschek, D., Huang, M. X., and Bulling, A. (2019). Reducing calibration drift in mobile eye trackers by exploiting mobile phone usage. In *Proc. International Symposium on Eye Tracking Research and Applications (ETRA)*.

Nakamura, K., Watanabe, T., and Jindai, M. (2013). Development of nodding detection system based on active appearance model. *Int. Symp. on System Integration*.

Naqvi, R., Arsalan, M., Batchuluun, G., Yoon, H., Kang, R., and Park (2018). Deep learning-based gaze detection system for automobile drivers using a nir camera sensor. *Sensors*, 18.

Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA. IEEE Computer Society.

Nguyen, L., Odobez, J., and Gatica-Perez, D. (2012). Using self-context for multimodal detection of head nods in face-to-face interactions. *Proc. of the 14th ACM Int. Conf. on Multimodal Interactions*.

Noris, B., Keller, J.-B., and Billard, A. (2011). A wearable gaze tracking system for children in unconstrained environments. *Computer Vision and Image Understanding*, 115(4):476–486.

Odobez, J. and Bouthemy, P. (1995). Robust multiresolution estimation of parametric motion models. *Journal of visual communication and image representation*, 6:348–365.

Oertel, C., Funes, K., Sheikhi, S., Odobez, J., and Gustafson, J. (2014). Who will get the grant? *Int. Conf. on Multimodal Interaction Workshop (UMMI)*.

Oertel, C., Lopes, J., Yu, Y., Funes, K., Gustafson, J., Black, A., and Odobez, J.-M. (2016). Towards building an attentive artificial listener: On the perception of attentiveness in audio-visual feedback tokens. In *18th ACM Int. Conf. on Multimodal Interaction (ICMI)*.

Papazov, C., Marks, T. K., and Jones, M. (2015). Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features. In *CVPR*.

Park, S., Mello, S. D., Molchanov, P., Iqbal, U., Hilliges, O., and Kautz, J. (2019). Few-shot adaptive gaze estimation. In *arXiv*.

Park, S., Spurr, A., and Hilliges, O. (2018). Deep Pictorial Gaze Estimation. In *European Conference on Computer Vision (ECCV)*, pages 741–757.

Park, S.-Y. and Subbarao, M. (2003). An accurate and fast point-to-plane registration technique. *Pattern Recogn. Lett.*, 24(16):2967–2976.

Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009). A 3d face model for pose and illumination invariant face recognition. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, AVSS '09, pages 296–301, Washington, DC, USA. IEEE Computer Society.

Pumarola, A., Agudo, A., Martinez, A., Sanfeliu, A., and Moreno-Noguer, F. (2018). Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Qin, Y., Lien, K.-C., Turk, M., and Höllerer, T. (2015). Eye Gaze Correction with a Single Webcam Based on Eye-Replacement. In *International Symposium on Visual Computing (ISVC)*, pages 599–609.

Ranjan, R., Patel, V. M., and Chellappa, R. (2016). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *CoRR*, abs/1603.01249.

Ranjan, R., Sankaranarayanan, S., Castillo, C. D., and Chellappa, R. (2017). An all-in-one convolutional neural network for face analysis. In *12th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2017, Washington, DC, USA, May 30 - June 3, 2017*, pages 17–24.

Recasens, A., Khosla, A., Vondrick, C., and Torralba, A. (2015). Where are they looking? *Advances in Neural Information Processing Systems*, pages 199–207.

Rudenko, A., Kucner, T. P., Swaminathan, C. S., Chadalavada, R. T., Arras, K. O., and Lilienthal, A. J. (2019). ThÖr: Human-robot indoor navigation experiment and accurate motion trajectories dataset. In *arXiv*.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Shi, L., Copot, C., and Vanlanduit, S. (2019). What are you looking at? detecting human intention in gaze based human-robot interaction. In *arXiv*.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6.

Siegfried, R., Yu, Y., and Odobez, J.-M. (2019). A deep learning approach for robust head pose independent eye movements recognition from videos. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, pages 31:1–31:5, New York, NY, USA. ACM.

Smith, B. A., Yin, Q., Feiner, S. K., and Nayar, S. K. (2013). Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST)*, pages 271–280.

Sugano, Y., Matsushita, Y., and Sato, Y. (2014). Learning-by-synthesis for appearance-based 3d gaze estimation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1821–1828. IEEE.

Sun, Y., Wang, X., and Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *CVPR*.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *CVPR*.

Tan, K.-H., Kriegman, D. J., and Ahuja, N. (2002). Appearance-based eye gaze estimation. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 191–195. IEEE.

Tan, W. and Rong, G. (2003). A real-time head nod and shake detector using hmms. *Expert Systems with Applications*, 25:461–466.

Tewari, A., Zollhöfer, M., Kim, H., Garrido, P., Bernard, F., Pérez, P., and Theobalt, C. (2017). Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *arXiv*.

Thomas, D. and Taniguchi, R.-i. (2016). Augmented blendshapes for real-time simultaneous 3d head modeling and facial motion capture. In *CVPR*.

Timm, F. and Barth, E. (2011). Accurate eye centre localisation by means of gradients. In *Proceedings of the Int. Conference on Computer Theory and Applications (VISAPP)*, volume 1, pages 125–130, Algarve, Portugal. INSTICC.

Venkateswarlu, R. et al. (2003). Eye gaze estimation from a single image of one eye. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 136–143. IEEE.

Venturelli, M., Borghi, G., Vezzani, R., and Cucchiara, R. (2017). From depth data to head pose estimation: a siamese approach. *arXiv preprint arXiv:1703.03624*.

Vetter, T. and Blanz, V. (1998). Estimating Coloured 3D Face Models from Single Images: An Example Based Approach. In *ECCV*.

Vidal, M., Turner, J., Bulling, A., and Gellersen, H. (2012). Wearable eye tracking for mental health monitoring. *Computer Communications*, 35(11):1306–1311.

Villanueva, A., Ponz, V., Sesma-Sanchez, L., Ariz, M., Porta, S., and Cabeza, R. (2013). Hybrid method based on topography for robust detection of iris center and eye corners. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(4).

Wang, F., Han, H., Shan, S., and Chen, X. (2017). Deep multi-task learning for joint prediction of heterogeneous face attributes. In *12th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2017, Washington, DC, USA, May 30 - June 3, 2017*, pages 173–179.

Wang, K., Zhao, R., and Ji, Q. (2018). A hierarchical generative model for eye image synthesis and eye gaze estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei, H., Scanlon, P., Li, Y., Monaghan, D., and O'Connor, N. (2013). Real-time head nod and shake detection for continuous human affect recognition. *In Image Analysis for Multimedia Interactive Services workshop (WIAMIS)*.

Weiner, D. and Kiryati, N. (2002). Gaze redirection in face images. *IEEE Convention of Electrical and Electronics Engineers in Israel*, pages 78–80.

Weise, T., Bouaziz, S., Li, H., and Pauly, M. (2011). Realtime performance-based facial animation. *SIGGRAPH*.

Wetherby, A. M., Guthrie, W., Woods, J., Schatschneider, C., Holland, R. D., Morgan, L., and Lord, C. (2014). Parent-implemented social intervention for toddlers with autism: An rct. *Pediatrics*, 134(6):1084–1093.

Wolf, L., Freund, Z., and Avidan, S. (2010). An eye for an eye: A single camera gaze-replacement method. Technical report.

Wood, E., Baltruaitis, T., Zhang, X., Sugano, Y., Robinson, P., and Bulling, A. (2015). Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. *IEEE International Conference on Computer Vision (ICCV)*, pages 3756–3764.

Wood, E., Baltrušaitis, T., Morency, L. P., Robinson, P., and Bulling, A. (2016a). A 3D morphable eye region model for gaze estimation. *European Conference on Computer Vision (ECCV)*, pages 297–313.

Wood, E., Baltrušaitis, T., Morency, L.-P., Robinson, P., and Bulling, A. (2016b). Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 131–138.

Wood, E., Baltrušaitis, T., Morency, L. P., Robinson, P., and Bulling, A. (2018). Gazedirector: Fully articulated eye gaze redirection in video. *Eurographics*, pages 217–225.

Wood, E. and Bulling, A. (2014). Eyetab: Model-based gaze estimation on unmodified tablet computers. *ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 3–6.

Xiong, Y., Kim, H. J., and Singh, V. (2019). Mixed effects neural networks (menets) with applications to gaze estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, X. and Kakadiaris, I. A. (2017). Joint head pose estimation and face alignment framework using global and local cnn features. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 642–649.

Yim, J., Jung, H., Yoo, B., Choi, C., Park, D., and Kim, J. (2015). Rotating your face using multi-task deep neural network. In *CVPR*, pages 676–684. IEEE Computer Society.

Yu, Y., Funes Mora, K. A., and Odobez, J.-M. (2017). Robust and accurate 3d head pose estimation through 3dmm and online head model reconstruction. In *Face and Gesture*.

Yu, Y., Liu, G., and Odobez, J.-M. (2018a). Deep multitask gaze estimation with a constrained landmark-gaze model. *European Conference on Computer Vision Workshop (ECCVW)*.

Yu, Y., Liu, G., and Odobez, J.-M. (2019). Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yu, Y., Mora, K. A. F., and Odobez, J. (2018b). Headfusion: 360° head pose tracking combining 3d morphable model and 3d reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(11):2653–2667.

Zhang, X., Huang, M. X., Sugano, Y., and Bulling, A. (2018). Training person-specific gaze estimators from user interactions with multiple devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 624:1–624:12, New York, NY, USA. ACM.

Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. (2015). Appearance-based gaze estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520.

Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. (2016). It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. (2017). MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–14.

Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2014). Facial landmark detection by deep multi-task learning. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, pages 94–108.

Zhou, E., Fan, H., Cao, Z., Jiang, Y., and Yin, Q. (2013). Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCV Workshop*.

Zhu, J., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *International Conference on Computer Vision (ICCV)*.

Zhu, W. and Deng, H. (2017). Monocular free-head 3d gaze tracking with deep learning and geometry constraints. In *The IEEE International Conference on Computer Vision (ICCV)*.

Zhu, X., Lei, Z., Liu, X., Shi, H., and Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *CVPR*.

Zitnick, C. L., Gemmell, J., and Toyama, K. (1999). Manipulation of video eye gaze and head orientation for video teleconferencing. *Microsoft Research MSR-TR-99-46*.

Yu Yu

PhD Candidate, Department of Electrical Engineering
École polytechnique fédérale de Lausanne (EPFL), Switzerland

personal homepage rainyucool@gmail.com (+41) 767829539

EDUCATION

École polytechnique fédérale de Lausanne (EPFL) **Switzerland**
PhD Candidate in Electrical Engineering *07.2015 – 01.2020 (Expected)*
Interest: Gaze Estimation, Head Pose Estimation, Image Synthesis

Xi'an Jiaotong University **China**
M.S. in Pattern Recognition and Intelligence System, GPA: 87.6/100 (top 1%) *09.2011 – 07.2014*
Interest: Hand Gesture Recognition, Document Image Analysis

Xi'an Jiaotong University **China**
B.S. in Information Engineering, GPA: 89.6/100 (top 3%) *09.2007 – 07.2011*

EMPLOYMENT

Idiap Research Institute **Switzerland**
Research Assistant *07.2015 – Present*
Interest: Gaze Estimation, Head Pose Estimation, Image Synthesis

Alibaba Group **China**
Algorithm Engineer *07.2014 – 07.2015*
Interest: Data Mining, Natural Language Processing

INTERNSHIP

SenseTime **China**
Research Intern *02.2019 – 06.2019*
Interest: Person Search, Graph Neural Network

ACADEMIC PROJECTS

Gaze Estimation and Gaze Activity Recognition *03.2017 - Present*

- **Unsupervised Representation Learning for Gaze Estimation:** Implemented a framework for unsupervised gaze representation learning which relies on a gaze redirection network and uses the gaze representation difference of the input and target images as the redirection variable. The achieved gaze representation is of low dimension and has very clear physical meaning. It works for few-shot gaze estimation, cross-dataset gaze estimation, gaze network pretraining, and another task (head pose estimation).
- **Gaze Activity Recognition:** Implemented a deep learning method which directly processes the eye image video streams to classify them into eye fixation, eye saccade, and eye blink. This work differs from previous works which take gaze signal as input.
- **Few-shot User-Specific Gaze Adaptation via Gaze Redirection Synthesis:** Implemented a gaze redirection framework to synthesize user-specific eye samples and used these gaze redirection sample to improve user-specific gaze estimation. This work alleviates the data lacking problem to some extent and applies gaze redirection to gaze estimation for the first time to our best knowledge.
- **A Differential Approach for Gaze Estimation:** Implemented a novel differential method for gaze

estimation where a network is trained to predict the gaze differences between two eye images of the same subject (instead of the absolute gaze of one sample). This work takes calibration samples in testing time and addresses the problem of user-specific bias.

- **Low Resolution Eye Gaze Estimation based on Eye Region Segmentation:** Implemented a multi-task network which addresses two tasks, eye segmentation and eye colorization. The eye segmentation result is then used for geometric gaze estimation. This work combines colorization with segmentation for the first time to our best knowledge.
- **Multitask Gaze Estimation based on a Constrained Landmark-Gaze Model:** Introduced a Constrained Landmark-Gaze Model (CLGM) which models the joint variation of eye landmarks and gaze directions and implemented a network inferring the parameters of CLGM model. This work improves over state-of-the-art results in challenging free head pose gaze estimation tasks.

Graph Neural Networks with application to Person Search *02.2019 – 06.2019*

Head Pose Estimation and Head Gesture Recognition *07.2015 – 03.2017*

- **Head Pose Estimation based on 3D Morphable Model and Online 3D Head Reconstruction:** Implemented a framework which combines the strengths of a 3DMM model fitted online with a prior-free reconstruction of a 3D full head model providing support for pose estimation from any viewpoint. This work achieves accurate and robust performance in challenging scenarios like extreme head pose and fast head motion.
- **Head Gesture Recognition for Social Interaction Analysis:** Implemented a novel head nod detection approach based on a full 3D face centered rotation model where the head rotation dynamic is computed within the head coordinate. This work achieves robust nod detection performance in real and unconstrained setting.

INDUSTRIAL PROJECTS

Recommendation System Development *11.2014 - 07.2015*

- Mining potential customers for e-commerce.
- Mining items related to hotspot news.

Deep Learning on Large Scale Distributed System *07.2014 – 11.2014*

- Implementing deep belief nets on large scale distributed dystems (in a parameter server framework).

PUBLICATIONS

Journal Papers

- Gang Liu, **Yu Yu**, Kenneth Alberto Funes Mora and Jean-Marc Odobez, "A Differential Approach for Gaze Estimation", IEEE Transactions on Pattern Analysis and Machine Intelligence (**TPAMI**), accepted, 2019
- Gang Liu, Kenneth Alberto Funes Mora, **Yu Yu** and Jean-Marc Odobez, "Low Resolution Eye Image Segmentation for Geometric Gaze Estimation", IEEE Transactions on Image Processing (**TIP**), **submitted**, 2018
- **Yu Yu**, Kenneth Alberto Funes Mora and Jean-Marc Odobez, "HeadFusion: 360 Head Pose Tracking Combining 3D Morphable Model and 3D Reconstruction", IEEE Transactions on Pattern Analysis and Machine Intelligence (**TPAMI**), vol.40, issue 11, 2018

Conference Papers

- **Yu Yu**, Jean-Marc Odobez, "Unsupervised Representation Learning for Gaze Estimation", Arxiv 2019
- **Yu Yu**, Gang Liu and Jean-Marc Odobez, "Improving User-Specific Gaze Estimation via Gaze Redirection Synthesis", Conference on Computer Vision and Pattern Recognition (**CVPR**) 2019
- Remy Siegfried, **Yu Yu** and Jean-Marc Odobez, "A Deep Learning Approach for Robust Head Pose Independent Eye Movements Recognition from Videos", ACM Symposium on Eye Tracking Research Applications (**ETRA**) 2019
- Gang Liu, **Yu Yu**, Kenneth Alberto Funes Mora and Jean-Marc Odobez, "A Differential Approach for Gaze Estimation with Calibration", British Machine Vision Conference (**BMVC**) 2018
- **Yu Yu**, Gang Liu and Jean-Marc Odobez, "Deep Multitask Gaze Estimation with Constrained Landmark-Gaze Model", Workshop of European Conference on Computer Vision (**ECCV Workshop**) 2018
- Remy Siegfried, **Yu Yu** and Jean-Marc Odobez, "Towards the Use of Social Interaction Conventions as Prior for Gaze Model Adaptation", 19th ACM International Conference on Multimodal Interaction (**ICMI**) 2017
- **Yu Yu**, Kenneth Alberto Funes Mora and Jean-Marc Odobez, "Robust and Accurate 3D Head Pose Estimation through 3DMM and Online Head Model Reconstruction", 12th IEEE International Conference on Automatic Face and Gesture Recognition (**FG**) 2017
- Catharine Oertel, José David Lopes, **Yu Yu**, Kenneth Alberto Funes Mora, Joakim Gustafson, Alan Black and Jean-Marc Odobez, "Towards building an attentive artificial listener: on the perception of attentiveness in audio-visual feedback tokens", 18th ACM International Conference on Multimodal Interaction (**ICMI**), 2016
- Yiqiang Chen, **Yu Yu** and Jean-Marc Odobez, "Head Nod Detection from a Full 3D Model", Workshop of International Conference on Computer Vision (**ICCV Workshop**) 2015
- **Yu Yu**, Yonghong Song, Yuanlin Zhang, "Real Time Fingertip Detection with Kinect Depth Image Sequences", International Conference on Patten Recognition (**ICPR**) 2014.
- **Yu Yu**, Yonghong Song, Yuanlin Zhang, Shu Wen, "A Shadow Repair Approach for Kinect Depth Maps", Asian Conference on Computer Vision (**ACCV**) 2012
- Shu Wen, Yonghong Song, Yuanlin Zhang, **Yu Yu**, "A Phase-based Approach for Caption Detection in Videos", Asian Conference on Computer Vision (**ACCV**) 2012

PROFESSIONAL SERVICE

Reviewer: IJCV, ICCV 2019, AAAI 2020, CVPR 2020, ECCV 2020

SKILLS

Programming Languages
Python, C++, Java, Matlab

Deep Learning Tools
PyTorch

AWARDS AND ACHIEVEMENTS

Facebook Synthetic Eye Generation Challenge, 2nd Place *2019*

Idiap Annual Paper Award *2018*

Chinese National Scholarship *2010 and 2013*