Unconstrained Gaze Estimation Using RGB-D Camera

Amine Kacete

N° d'ordre : 2016-35-TH

CentraleSupélec

Ecole Doctorale MATISSE

« Mathématiques, Télécommunications, Informatique, Signal, Systèmes Electroniques »

Laboratoire IETR/FAST

THÈSE DE DOCTORAT

DOMAINE :
Spécialité :
Traitement du signal

Soutenue le
15 décembre 2016

par :

Amine Kacete

Unconstrained Gaze Estimation Using RGB-D Camera

Composition du jury :

Directeur de thèse :	Renaud Séguier	Professeur, CentraleSupelec
Co-directeur de thèse 1 :	Michel Collobert	Ingénieur de recherche, orange labs
Co-directeur de thèse 2:	Jérôme Royan	Architecte principale, IRT b-com
Président du jury :	Jean-Marc Odobez	Professeur, IDIAP, Suisse.
Rapporteurs :	Alice Caplier	Professeur, GIPSA-LAB, Grenoble
Examinateurs :	Kévin Bailly	Maître de conférence, ISIR-UPMC, Paris

Acknowledgments

The work achieved in this thesis is one of the most important chapters in my life. This experience would certainly not have come to its successful conclusion without the trust, help and support of colleagues, friends and family.

First and foremost, I would like to express my deep-felt gratitude to my supervisors, Prof. Renaud Séguier, Michel Collobert and Dr. Jérôme Royan for giving me the opportunity to work with IRT b<>com, I will be always grateful for their guidance, trust and for the freedom I was granted throughout these past three years. Their advice and support have had a great and positive impact on this work.

Second, it has been a privilege to work closely with my friend and colleague Jérémy Lacoche. His great experience and skills in computer graphic have allowed me to solve many problems related to rendering. I would also like to thank my colleagues from the Immersive Interaction team: Morgan, Thomas, David, Fabienne, Henry, Vincent, Cyndie and Nicole who all contributed to create and maintain a pleasant and simulating working environment. Special thanks go to Lucie Petta who accepted to proofread parts of this manuscript.

Moreover, I would like to take this opportunity to thank my valuable colleagues in CentralSupelec: Jérôme, Catherine and Caroline for their generosity in sharing comprehensive knowledge. I also want to thank Nicolas and Vincent from Dynamixyz for the many interesting and inspiring discussions we had.

I would like to warmly thank my family, specially my parents for their love, car and help despite the long distances we have faced.

Last but not least, I want to express a particular thanks to my dear friend Joseph, for his positive attitude, great friendship and constant support.

Abstract

In this work, we tackled the automatic gaze estimation problem in unconstrained user environments. By using RGB-D sensor, we aimed to exploit the multi-modal data to provide a sufficiently accurate and robust system. This work can be summarized through 3 principle axes: model, paradigm and data.

In the first part of this thesis, we described in details Random Forest algorithm. Through our investigation, we formulated some tasks as a learning problems, we used decision forest as model to capture the mapping functions. We gave a global overview of this tool, and compared it to some machine learning techniques under different tasks. We finished this part by highlighting the recent achieved improvements of this algorithm in computer vision and medical image analysis. Through this survey, we reported some empirical proofs of the potential of Random Forest in handling highly non linear problems such as gaze estimation.

The second axis of this work is about gaze estimation paradigms. We first developed two automatic gaze estimation systems following two classical approaches: feature and semi appearance-based approaches. Our feature-based system is based on a robust eye pupil localization component which allows to build a 3D eye model. Combined with head pose estimation, a 3D gaze information can be inferred. The second system is fundamentally based on building a frontal gaze manifold corrected with the head pose parameters. This system aims to learn gaze information from eye image appearance under frontal configurations then uses head pose-based geometric transformation to infer the final gaze information. The major limitation of such paradigms lies in their way of designing gaze systems which assume a total independence between eye appearance and head pose blocks. To overcome this limitation, we converged to a novel paradigm which aims at unifying the two previous component and building a global gaze manifold. To achieve such unification, we built an input data from both RGB cue related to eye appearance, and depth cue related to head pose. A robust mapping between such input space and gaze information space is learned robustly. We performed a comprehensive comparisons between these systems under unconstrained environment and reported a deepen analysis about the obtained results.

The final axis of this work represents the data. Providing sufficient input data to learn mapping functions with a high ability of generalization is fundamental. We explored two global approaches across the experiments by using synthetic and real RGB-D gaze samples. For each type of data, we described the acquisition protocol

and evaluated the ability of handling the task in hand. We finished by performing a synthetic/real learning comparison in terms of robustness and accuracy and inferred some empirical correlations.

Résumé

Dans ce travail , nous avons abordé le problème de l'estimation automatique du regard dans des environnements utilisateur sans contraintes. L'estimation du regard joue un rôle dans plusieurs applications en vision par ordinateur spécialement dans l'analyse du visage. Dans la reconnaissance d'expressions faciales, il peut remonter une information très parlante sur l'état expressif et cognitif de la personne. Les interactions homme-machine IHM utilisent cette information comme métaphore principale de communication. Le regard peut être exploité dans des thématiques de contrôle et de surveillance, plusieurs constructeurs automobiles intègrent cette technologie pour vérifier l'état de fatigue du conducteur. Récemment, certaines recherches s'orientent vers l'utilisation du regard pour comprendre le comportement des clients dans les magasins, les données récoltées permettent d'élaborer des stratégies marketing de plus en plus intelligentes.

Plusieurs solutions industrielles sont aujourd'hui commercialisées et donnent des estimations précises du regard. Certaines ont des spécifications matérielles très complexes (des caméras embarquées sur un casque ou sur des lunettes qui filment le mouvement des yeux) et présentent un niveau d'intrusivité important. Ces solutions sont souvent non accessibles au grand public. D'autres utilisent un champ de caméras infra-rouge et se basent intégralement sur le reflet cornéen pour estimer le regard, seulement leur robustesse est fortement conditionnée par les conditions d'éclairage. Récemment, deux approches ont émergé dans l'estimation du regard, basée-caractéristique et basée apparence respectivement. Ces approches essaient à la fois, de réduire le niveau d'intrusivité dans le but de fournir plus de mobilité à l'utilisateur, et d'augmenter la robustesse de l'estimation. La première approche considère le regard comme un vecteur résultant d'un modèle géométrique de l'œil. Pour calibrer ce modèle pour chaque utilisateur, une étape d'extraction automatique de points caractéristiques autour de l'œil et une estimation de la pose de la tête est fondamentale. Pour s'affranchir de la calibration par utilisateur, la deuxième approche modélise l'estimation comme un problème de régression. L'idée est de trouver une fonction mapping suffisamment précise entre l'espace d'entrée représenté par l'image apparence de l'œil, et l'espace de sortie représenté par le vecteur regard. Basés sur ces deux dernières approches, les systèmes d'estimation automatique du regard ont réalisé une avancée considérable en termes de précision et de robustesse. Néanmoins, plusieurs verrous restent non résolus. Ce travail vise à produire un systéme d'estimation automatique du regard capable d'augmenter la liberté de mouvement

de l'utilisateur par rapport à la caméra (mouvement de la tête, distance par rapport au capteur), et de réduire la complexité du système en utilisant des capteurs relativement simples et accessibles au grand public.

Dans le premier axe de cette thèse, nous nous sommes focalisés sur un algorithme d'apprentissage automatique basé sur les champs aléatoires. En effet, au cours de nos études, nous avons formulé certaines tâches comme des problèmes de régression. Nous avons utilisé les arbres de décisions comme modèle pour apprendre les fonctions mapping. Nous avons commencé par donner un aperçu global de cet outil en détaillant l'aspect mathématique permettant de comprendre sa flexibilité à résoudre différentes problématiques d'apprentissages (classification, régression, estimation de densité ainsi que la réduction de dimension). Dans un deuxième temps, nous avons réalisé une étude comparative étendue sur plusieurs problématiques des champs aléatoires par rapport à d'autres algorithmes d'apprentissage (Machines à vecteurs support, Processus Gaussiens..). Cette étude nous a permis de ressortir rigoureusement les avantages de cette technique et de se prononcer objectivement sur sa capacité à être projeté sur nos problématiques. Pour conclure cette partie, nous avons mis en évidence les améliorations récentes apportées à cet algorithme en vision par ordinateur et en analyse d'imagerie médicale. Cette analyse souligne certaines limitations de l'algorithme original induisant des corrections de plus en plus sophistiquées.

Le deuxième axe présente les paradigmes d'estimation du regard. Dans un premier temps, Nous avons mis au point deux systèmes basés sur deux approches classiques: le premier basé caractéristiques et le deuxième basé semi-apparence. Nous avons introduit pour la première fois cette notion de semi-apparence pour désigner un système utilisant partiellement l'apparence pour estimer le regard. Notre système basé caractéristiques repose sur une composante robuste de localisation de la pupille permettant de construire un modèle géométrique de l'œil. Cette composante est le résultat d'un apprentissage en utilisant les champs aléatoires permettant de construire un espace de vote global sur la position de la pupille. En fournissant un corpus de données labélisé en position et classe de la pupille, les arbres sauvegardent des informations à la fois de classification et de régression permettant de construire un espace dit de Hough encodant toutes les hypothèses plausibles de la position de la pupille. Combinée à une autre composante qui est l'estimation de la pose de la tête par transformations géométriques, l'information regard final peut être calculée. Notre module de pose de la tête est basé également sur un apprentissage par champs aléatoires sur des images RVB-P labélisées. Notre second système repose principalement sur un paradigme semi-apparence. Ce paradigme vise à apprendre l'information regard à partir d'images d'apparences de l'œil dans une configuration exclusivement frontale. Pour construire un espace regard frontal, une étape de normalisation de pose permettant de construire une image frontale du visage de l'utilisateur est appliquée.

Pour estimer le regard final, la prédiction dite frontale est convoluée aux paramètres de la pose de la tête de facon géométrique. Dans notre système, nous avons détaillé notre méthode de normalisation de pose tout en illustrant son importance dans la

détection du visage sur deux différentes bases de données. Nous avons conclu cet axe par des comparaisons de ces deux systèmes par rapport à l'état de l'art ainsi qu'une comparaison directe de ces deux paradigmes. Cette analyse nous a permis de ressortir un inconvénient majeur de ces paradigmes résidant dans la conception des systèmes d'estimation du regard qui supposent une indépendance totale entre l'image d'apparence de l'œil et la pose de la tête. Ainsi, nous avons convergé vers un nouveau paradigme qui vise à unifier les composantes précédentes et construire un espace global du regard. Pour parvenir à une telle unification, nous avons modélisé le vecteur d'entrée comme une combinaison d'informations RVB liées à l'apparence des yeux, et de profondeur liée à la pose. Une fonction mapping entre cet espace d'entrée et l'espace regard est apprise.

Dans le dernier axe de cette thèse, nous avons développé notre dernier système d'estimation automatique du regard. Ce système repose sur un nouveau paradigme qui est intégralement apparence. Cette appellation est choisie pour marquer davantage la différence avec les systèmes décrits précédemment. En effet, l'axiome principal de ce paradigme est de coupler intégralement la composante de l'apparence de l'œil avec l'estimation de pose de la tête. Concrètement, nous exploitons la multimodalité de la Kinect pour extraire des informations RVB autour des yeux, ainsi qu'une information de profondeur autour du visage. Cette extraction est appliquée après une étape de détection du visage. Cette dernière nous permet de construire des données multicanaux (deux canaux RVB-yeux et un canal Profondeur-visage) encapsulant à la fois des informations regard et pose de la tête. Nous utilisons encore une fois les champs aléatoires pour apprendre le passage entre l'espace d'entrée multicanaux de très grande dimension et l'espace de sortie du regard de faible dimension. Dans cet axe, nous mettons en évidence l'importance des données d'apprentissage et leur influence sur la qualité de la prédiction finale. Pour entrainer des arbres de décision suffisamment robustes dotés d'une grande capacité de généralisation face à des scénarios très différents de l'apprentissage, il faut impérativement s'assurer de la pertinence des données d'apprentissage en termes de nombre et de variabilité. Malheureusement, aucune base de données regard existante ne répond à ces exigences. Aussi, nous avons décidé de construire nos propres données garantissant aux champs aléatoires une grande généralisation pendant la prédiction. Nous avons exploré deux principales approches à travers les expériences, en utilisant respectivement des échantillons RVB-P synthétiques et réels. Pour construire une base de données regard synthétiques, nous avons utilisé un modèle 3D de visage statistique déformable. Ce modèle a été construit à partir d'une analyse en composante principale de scans de plusieurs personnes de différents âges, l'ACP permet de ressortir des modes de variations contrôlant à la fois la forme et la texture du modèle induisant une nouvelle identité synthétique. Nous avons ajouté à ce modèle un mode de variation supplémentaire relatif au regard. Pour se faire, nous avons intégré un modèle 3D paramétrable d'yeux. Ce modèle comporte deux sphères texturées et un ensemble de points autour de l'œil. Le paramétrage est défini empiriquement et permet de contrôler le mouvement des yeux et des paupières produisant une labélisation automatique du regard. D'un autre côté, pour construire une base de

données réelle, nous avons suivi un protocole de captation rigoureux. Plusieurs personnes ont été conviées à suivre un point mobile affiché sur un écran intégralement calibré par rapport à la Kinect permettant de construire une vérité terrain regard suffisamment précise. Plusieurs configurations ont été mises au point (assis/debout, proche/loin du capteur) pour s'assurer de la variabilité des données RVB-P. Un premier apprentissage global a été réalisé en utilisant exclusivement des données synthétiques. Cet apprentissage a été testé dans des situations réelles avec de fortes contraintes de pose de la tête et de distance par rapport au capteur. Une autre expérimentation a été menée pour évaluer la pertinence de chaque canal dans la prédiction, et plus particulièrement la profondeur. Pour valider la robustesse du paradigme intégralement apparence vis-à-vis du semi-apparence, une étude basée sur des données synthétiques a été conduite comparant la précision de la prédiction. Pour finir cet axe, une derniére analyse a été réalisée dans une optique de comparaison d'apprentissage réel/synthétique. Cette expérience nous a permis de faire ressortir empiriquement un rapport de nombre de données d'apprentissage synthétiquse/réelles permettant de produire, approximativement, la même estimation du regard en termes de précision.

Nous avons conclu notre travail par une comparaison globale de nos trois systèmes d'estimation automatique du regard dans des environnements utilisateur différents. Cette comparaison montre les avantages et inconvénients de chaque système permettant d'émettre des perspectives sur trois volets. Le premier volet concerne directement la méthode d'apprentissage utilisée, plusieurs améliorations peuvent être envisagées pour booster d'une part sa discrimination et d'autre sa capacité à encapsuler de l'information sémantique. Le deuxième se projette sur les données synthétiques dans une optique d'amélioration du rendu et de réalisme des données d'apprentissage. En effet, d'autres protocoles peuvent être conçus pour mieux synthétiser les échantillons regard RVB-P. Une dernière perspective porte sur la construction d'une base de données RVB-P regard réelle, certaines réflexions peuvent être conduites dans le but d'améliorer le protocole d'acquisition ainsi que la qualité de labéllisation des données.

Ce travail a donné naissance à une autre collaboration entre le laboratoire Interactions Immersives de l'IRT b<>com et l'équipe FAST de CentraleSupelec. Cette collaboration se concrétise par une thèse reprenant les résultats de ce travail dans un autre cadre applicatif.

Contents

Chapter 1

Introduction

Chapter 2

Automatic gaze estimation: state-of-the-art

Contents

As described previously, the automatic gaze estimation systems aim to infer the 3D visual axis or the point of regard. This last decade, two main methodologies have been investigated by several researches, namely feature-based methods and appearance-based methods. A very comprehensive survey is described by [Hansen 2010].

2.1 Feature-based methods

These methods use geometrical assumptions to infer the gaze information. To this end, they build a person-specific geometrical model using a set of discriminative and invariant features around the eye image such as the pupil. This model is usually described as the eyeball geometry as illustrated in Fig.2.2 where a direct mapping from the local features locations to the gaze point is established. Since the local features of the eyes are very specific varying from one user to an other, a calibration session is required to determine the parameters of the model by collecting a set of gaze estimation samples.

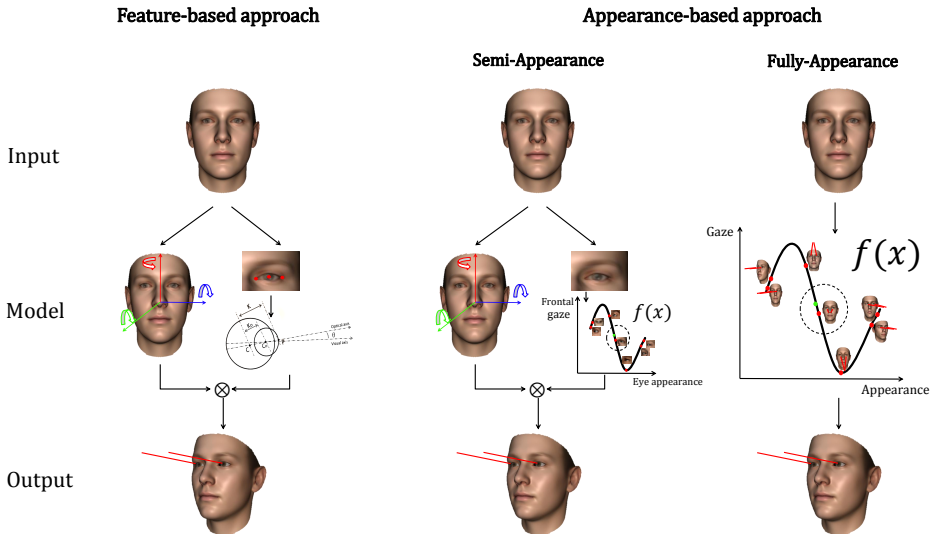


Figure 2.1: The two principal automatic gaze estimation approaches. (a) Feature-based methods: combining eye features locations and head pose parameters, the gaze information can be calculated. (b) Appearance-based methods: two paradigms are possible, learning gaze based on the eye appearance in frontal configuration, then correct the final estimation using head pose parameters. The second paradigm consists in learning gaze information using the full eye and face cues as unified input.

To achieve a 3D gaze estimation, these methods use the head pose parameters to project the 2D estimation in the world coordinates system. The head pose is conventionally defined with two global parameters, R and T, representing the rotation and translation of the user head respectively. Estimating accurately head pose parameters enhances significantly automatic gaze estimation system efficiency. The Wollaston illusion represented in Fig.2.3 illustrates the discriminative characteristic of these components. Here, we describe a comprehensive surveys related to head pose estimation and eye pupil localization methods reported these last years. Then we describe gaze estimation systems combining these two components.

2.1.1 Head pose estimation

As mentioned previously, head pose estimation is a fundamental component for the majority of automatic gaze estimation systems. A strong correlation exists between the gaze direction and the head orientation, measuring accurately this information can reduce the high-dimensionality of gaze estimation problem.

Head pose estimation has a variety of interpretations. In the context of computer vision, head pose estimation is most commonly interpreted as the ability to infer the orientation of a person's head relative to the view of a camera which is represented by the full 3D orientation and translations. A visual representation is represented in Fig.2.4

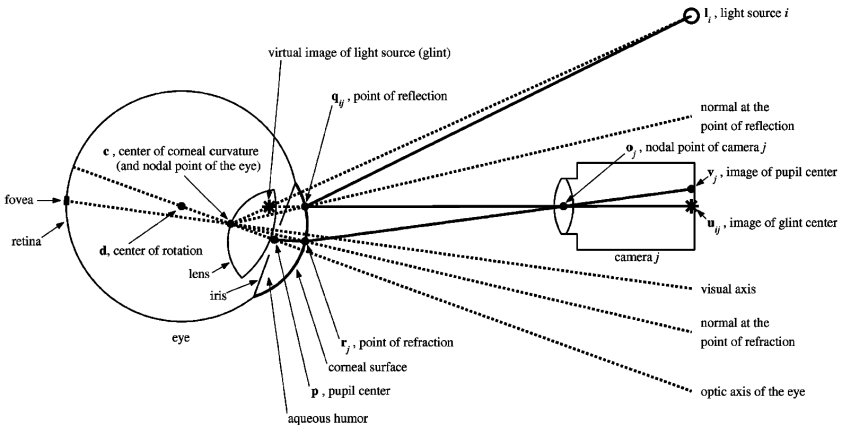


Figure 2.2: Geometric eyeball model (extracted from [Guestrin 2006]).

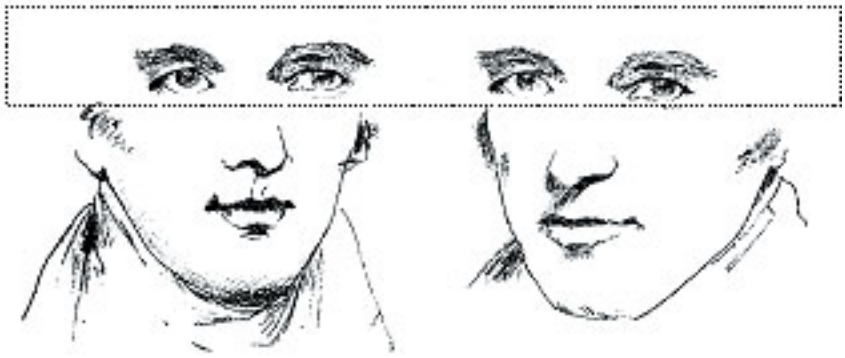


Figure 2.3: Wollatson illusion: the appearance eye images are exactly the same. By considering the head pose changes, the perception of the gaze direction is different.

Several methods for automatic head pose estimation are proposed in the literature. With respect to the survey given in [Murphy-Chutorian 2009], we classify these methods according to the global approach used and complete with more recent methods. Fig.2.5 illustrates a visual representation of some head pose approaches in computer vision this last decade.

- **Appearance template approach** This method is considered as the most intuitive approach. A set of sparse head pose annotated images is collected and used as templates. To infer the head pose, the RGB image test is compared simultaneously to the defined template using a similarity measurement. [Beymer 1994] used cross-similarity with a multi-scale strategy to compute the similarity, [Niyogi 1996] performed a standard mean square error. The major limitation of this approach remains in the discretization of the head pose manifold.

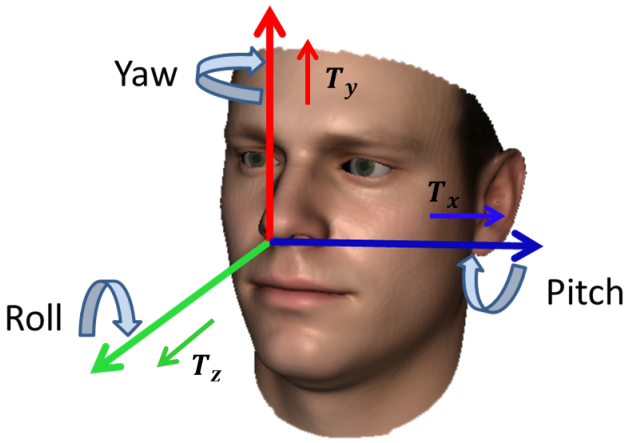


Figure 2.4: Head pose parameters. yaw, pitch and roll describe the orientation of the head (usually defined as the Euler angles). T_x, T_y and T_z represent the translation to the sensor which correspond to the head gravity center.

- **Detector array approach** This approach aims to train a single classifier for each specific head pose orientation. At the test step, the learned set of classifiers apply simultaneously to the image test, and the estimated head pose is retrieved by the classifier with the highest score. The main weakness of this approach is the same as the previous one. The works from [Rowley 1998], [Huang 1998] and [Jones 2003] are based on this approach.

- **Nonlinear approach** By using a supervised strategy, this approach learns the high non-linearity between the head appearances space and the pose parameters space. Unlike previous approaches, these methods gives a continuous estimation. Nevertheless, to achieve a sufficient generalization, a very representative training set is required. Different machine learning techniques are used to tackle this problem such as Support Vector Regression (SVR) in [Li 2000], Neural Networks in [Bruske 1998], [Zhao 2002], [Stiefelhagen 2004] and [Voit 2005], Particle Swarm in Optimization [Padeleris 2012], Random Forest in [Fanelli 2011], Deep Neural Networks in [Ahn 2014] and K-Nearest Neighbor with a triangular surface patch descriptor in [Papazov 2015].

- **Manifold learning approach** Instead of learning the mapping between the input and output spaces, this approach aims to lie the samples on a low-dimensional continuous space. Different dimensionality reduction methods are used to learn the mapping from the head pose input images space and the new low dimensional manifold, such as principle component analysis (PCA) [McKenna 1998], [Sherrah 1999] and linear discriminative analysis (LDA) [Chen 2003], [Wu 2008]. The fact that the assumption that head pose is mainly the only variable responsible for dimensionality reduction is not

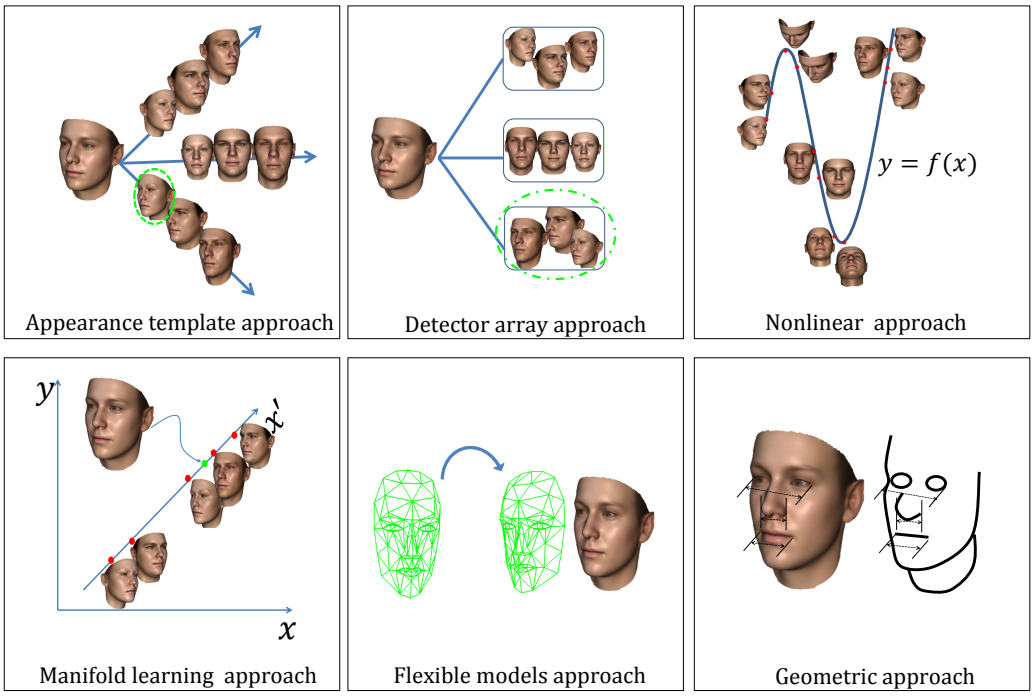


Figure 2.5: A comprehensive illustration of the existing approaches of human head pose estimation.

guaranteed, it represents a real weakness.

- **Generic deformable models approach** These methods rely on fitting a non-rigid model to the test image of the facial structure. The fitting consists in minimizing a cost function which evaluates the similarities between the test data and the current instance generated by the model. The head pose is estimated by comparing the current parameters of the model to a set of predefined head pose configurations. The parameters of the models are learned from a set of training data of the facial structure configurations. The most representative approach using generic models are [Krüger 1997] and [Hu 2004]. Achieving a robust and accurate fitting is a challenging task which is directly linked to the generalization of the generic model and the robustness in facial feature localization.

- **Geometric approach** Inspired by the human perception of the head pose [Wilson 2000], these methods are based on the localization of some key facial cues. The head pose is determined by minimizing the distances between the projection of the 3D facial landmarks of 3 3D predefined model and the estimated landmarks. The fact that the head pose estimation is directly linked to the localization of the landmarks which is highly constrained by the resolution imaging and the

facial expressing, makes achieving robustness across these conditions challenging. Among the representative methods we can cite [Horprasert 1997] [Xiong 2005] [Canton-Ferrer

- **Tracking approach** This approach considers the temporal information. By measuring the change of head using temporal continuity and smooth motion strategy, the head pose can be estimated. This approach presents a high accuracy compared to the previous methods but remains strongly constrained by the estimation initialization namely the frontal configuration. Several methods have used this strategy these last years. The tracking can be based on facial features localization [Yang 2002] [Jang 2008] [Wang 2012], optical flow [Morency 2002], particle filters [Oka 2005], rigid 3D model [La Cascia 2000] and [Lefèvre 2009] and nonrigid 3D model, [Amberg 2008], [Weise 2011], [Papazov 2015].

- **Hybrid approach** To overcome the limitations of the previous methods, this approach combines different strategies. Nonlinear mapping and reduction dimension in [Huang 2004], manifold learning with flexible model in [Wu 2008], tracking with geometrical assumptions [Heinzmann 1998] [Newman 2000],and finally tracking and dimensionality reduction [Baltruvsaitis 2012], [Morency 2003] [Morency 2010].

We perform a comprehensive comparison of these methods as done in [Murphy-Chutorian 2009] in Tab.2.1. We complete this table with some recent approaches as follows:

Methods	Mean Absolute Error			Classification	Number of discrete
	Yaw	Pitch	Roll	accuracy	poses
[Beymer 1994]	21.2°	5.2°	-	-	
[Krüger 1997]	-	-	-	92.0%	5
[Stiefelhagen 2004]	9.5°	9.7°	-	-	-
[Voit 2005]	8.5°	12.5°	-	-	-
[Wu 2008]	-	-	-	75.4%	86
[Lefèvre 2009]	4.4°	3.3°	2.0°	-	-
[Fanelli 2011]	5.7°	5.1°	-	-	-
[Padeleris 2012]	1.62°	2.05°	-	-	-
[Baltruvsaitis 2012]	6.29°	5.10°	11.29°	-	-
[Mora 2012]	4.53°	2.76°	3.95°	-	-
[Ahn 2014]	2.8°	3.4°	2.6°	-	-
[Papazov 2015]	3.8°	3.5°	5.4°	-	-

Table 2.1: Mean gaze estimation error across two user-sensor distances.

According to the results reported in Tab.2.1, the classification [Krüger 1997] and [Wu 2008] methods are not suitable for our task since we consider a continuous user gaze space. [Lefèvre 2009] achieved promising results but still handles relatively small distances. [Padeleris 2012] reported better results on large distances, nevertheless their method needs important computational time (they enhanced processing time by using GPU's architecture). [Mora 2012] and [Baltruvsaitis 2012] presented

real-time estimation with sufficient accuracy but need strong initialization assumption. [Ahn 2014] and [Papazov 2015] achieved robustly good results by using heavy complex architectures. By using Random Forest algorithm, the method presented by [Fanelli 2011] represents a very interesting approach since it offers a good balance between robustness, accuracy and runtime.

2.1.2 Eye pupil localization

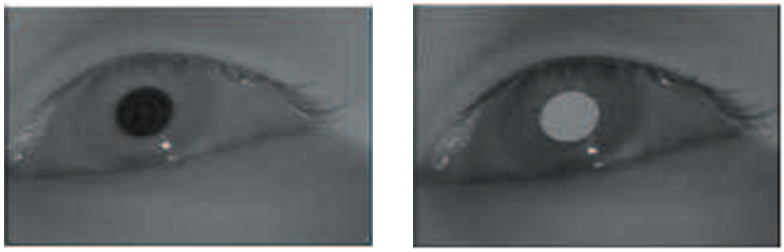
Eye pupil location plays a key role in feature-based automatic gaze estimation systems and their movements give an important information in many applications mentioned previously such as cognitive and psychological processes.

Two global existing methods can conveniently be distinguished by the type of data they rely on, infrared (IR) or visual (RGB) images. Here we describe the most relevant methods reporting state-of-the-art results for each category.

2.1.2.1 IR-based methods

The majority of the commercial eye-tracking systems use infrared light (IR) (with wavelength around $780 - 880nm$) to estimate the eye-pupil localization. The main idea is about exploiting the light reflexion with regards to the light source location, the pupil presents different behavior, as illustrated in Fig.2.6. The pupil will be bright if the light source location is close to the optical axis and dark otherwise, making its segmentation and tracking simple. The main advantages of this approach are efficiency and simplicity, [Morimoto 2000], [Ji 2002] and [Hansen 2007] used the differences between the dark and bright pupil images acquired from IR sources synchronized with an RGB camera to produce a robust tracking. To reduce the problem of the reflexion of IR light sources on glasses [Ebisawa 1998] proposed a pupil brightness stabilization process. [Haro 2000] combined eye appearance with the bright pupil effect to overcome the illumination variation and distinguish the pupil from the objects with the same brightness present in the background. [Zhu 2002] proposed a real-time pupil tracking using Support Vector Machine (SVM) to train a robust classifier for pupil blobs, the final pupil is detected by refinement using a tracking process based on Kalman filtering and mean-shift clustering.

These methods achieved remarkable results on pupil tracking, however, many challenging conditions are still unsolved such as illumination condition (constrained by the indoor scenarios and low lighting), head pose variation (brightness of the pupil is strongly affected by the head movements). The most common solution used in the automatic gaze estimation systems based on IR lighting is to setup the sensors in an embedded way as in the head mounted systems discussed previously. This solution, nevertheless, represents a real intrusiveness. These last years, IR-based methods are less and less popular giving way to the RGB-based methods.



(a) (b)

Figure 2.6: IR imaging. (a) Illustration of a typical dark pupil when the light source is away from the optical axis. (b) The light source is close to the optical axis making a bright pupil.

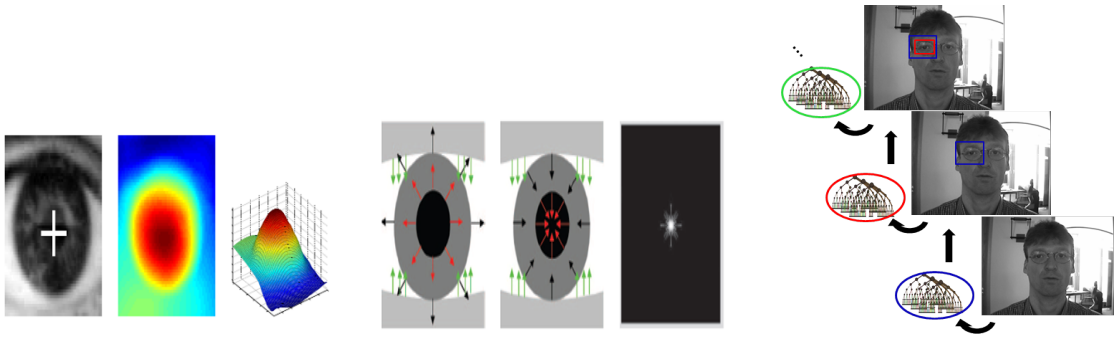
2.1.2.2 RGB-based methods

Taking an RGB image as input, the pipeline of these methods consists in detecting the face using the method from [Viola 2001], which extracts rough regions around the eyes using anthropomorphic relations then estimates the spatial position of the pupil on the image space. We describe below the most relevant methods from the-state-of-the-art.

- **Means of Gradients:** The method from [Timm 2011] uses the geometric aspect of the pupil by defining an objective function based on an image gradient that takes its maximum at the intersection of the gradient vectors. This method is very robust under illumination and scale variations. Nevertheless, with significant head pose variations, circularity of the pupil is not guaranteed giving bad estimates.

- **Curvature of Isophotes:** Represents a set of curves that connect points of equal intensity. The method from [Valenti 2008] uses these points to identify each candidate location and extracts SIFT vector [Lowe 1999] and compares it to a given template in a defined database to get the final decision. Like the previous method, this solution suffers significantly from head pose variations since vectors pointing to the isophotes centers give wrong estimates.

- **Randomized Trees** The method described in [Markuš 2014] ignores geometrical assumptions. Instead, the authors trained in a cascaded way ensembles of trees to learn the mapping between eyes images appearances and 2D pupil locations. Each ensemble processes a given scale i and represents the input of the following ensemble relative to the scale $i-1$ up to the final output (the number of scales defines the number of ensembles). Their final learning includes one hundred trees organized in five ensembles trained with six million images. This method seems to be the most robust and accurate approach under different constraints such as low resolution, illumination conditions and head pose variations but it needs a strong initialization assumption due to



(a) (b) (c)

Figure 2.7: State-of-the-art eye pupil localization. (*a*) Visualizing the objective function used in [Timm 2011] to find the mean of gradients and its maximum corresponding to the final estimation. (*b*) Some potential candidates vectors pointing to the isophotes centers and their aggregation used in the method from [Valenti 2008]. (*c*) [Markuš 2014] describes the cascade strategy used to refine the eye-pupil localization.

the designing of their processing model and it suffers from some intra-user variations.

Fig.2.7 summarizes these state-of-the-art methods, we notice again, that the approach based on Random Forest decision presented by [Markuš 2014] achieved a high estimation accuracy compared to the baseline methods.

2.1.3 Geometric gaze estimation

Many existing methods based on the components discussed previously can be reported these last years. [Wang 2002] and [Kohlbecher 2008] inferred the gaze information using the shape of the pupil estimated through an elliptic fitting. In addition to the pupil location information, [Ishikawa 2004] uses the eye corners locations estimated through an AAM [Cootes 2001] fitting. By combining these two information, the center and the radius of the eye ball are estimated giving the two angles of the visual axis. [Matsumoto 2000] proposed a 3D information of the eye ball to estimate the 3D visual axis provided by a stereo setup, the same strategy is performed by [Chen 2008] with a single camera by adding a calibration step. [Valenti 2012] used a 3D predefined cylindrical head model with a tracking strategy to compute the head pose rigid parameters and estimated the eye pupil localization using the method described in [Valenti 2008] with a calibration plane inside the head field of view. [Bär 2012] and [Jianfeng 2014] estimated gaze using a depth sensor, the first method used a multi-template ICP algorithm to estimate the head pose and a template matching approach based on an elliptical fitting to fix the eyeball parameters. The second method used a flexible model fitting approach to estimate the head pose parameters, coupled to a pupil detection and a calibration step by gazing a fixed 3D points, the visual axis is calculated.

[Morimoto 2005], [Guestrin 2006] and [Zhu 2007] used the corneal reflexion information based on one or multiple IR light sources. The relative position of the eye pupil to the corneal reflexion is exploited in a geometrical way to calculate the gaze information. Usually these methods operate in a 2D screen scenario relying on determining the gazed 2D point, without introducing head pose informations.

On one hand, these recent years, the eye-tracking technology tends more and more towards a wide consumer market uncontrolled environments, on the other hand, the feature-based methods still requires a high resolution imaging and a very heavy and constrained calibration process, allowing more interest to appearance-based methods.

2.2 Appearance-based methods

Unlike feature-based methods which rely on performing eye features explicit extraction and assuming prior geometrical assumptions, these approaches aim to learn a direct mapping from the high dimensional eye images to the low dimensional space of the gaze information, as illustrated in Fig.2.1. [Baluja 1994] and [Xu 1998] trained a neural network using $2k$ and $4k$ labeled training samples respectively. [Tan 2002] collected 252 training samples to build a manifold using the local linearity of the eye appearance and estimated an unknown testing sample using a linear interpolation. [Hansen 2002] exploited the Markov model interpolation to enhance the generalization across unseen scenarios such as gaze sample under head movement. [Williams 2006] introduced a Sparse Semi-Supervised Gaussian Process S^3GP completing the labeled training data with unlabeled samples. Due to the fact that the approach is fully Bayesian, the estimation is given with an uncertainty measurement for an unknown test gaze sample. [Sugano 2008] proposed an incremental learning strategy using an on-line sample acquisition from a video stream updating the mapping function for a number of limited head pose configurations. [Sugano 2010] proposed a visual saliency map-based strategy to generate the training data through a video stream. The saliency maps are considered as the probability distribution of gaze points for a specific user, the function mapping is established using a Gaussian Process regression. To further reduce the number of training samples, [Lu 2011b] introduced the adaptive linear to learn the mapping function on a very sparse training set. To decrease the non linearities of the mapping function due to head pose movements which introduce a very representative distortion in the appearance of the eye images, [Lu 2011a] proposed to compensate adaptive linear regression mapping under frontal scenarios with a geometrical calculation. Using the same paradigm as the last method, [Mora 2012] projected the training gaze sample in a frontal configuration manifold and then applied a regression mapping. This method seems the closest to our approach, we will further discuss in details in the next chapters. Very recently [Zhang 2015] introduced a deep learning strategy using convolutional neuronal network to learn the mapping function on a representative set of gaze training samples from very unconstrained scenarios.

One of the most important limitation of the majority of these previous methods is the assumption of the fixed head configuration when the human gazing process in naturally under head movements, which decrease drastically the estimation accuracy (error jumps from 4° on frontal configuration to more than 10° under head pose changes scenarios). [Mora 2012] and [Lu 2011a] achieved better results (error less than 3°) in such scenarios and by considering that the head pose component is fully decoupled from the gaze training samples manifold, the final gaze estimation accuracy is strongly linked to head pose estimation accuracy which depends on the user-sensor distance, so these methods rely on a relatively low user-sensor distances. [Zhang 2015] performed an acceptable gaze estimation accuracy under these conditions, however, this method needs an important computational time in the testing step and is still not real-time. In addition to these limitations, the number and nature of training samples required to achieve a good generalization of the mapping function under unconstrained environment are frequently encountered.

Moreover, all these described methods treat the gaze estimation with a strong assumption of the head-eye blocks independence. We decided to define them as a semi appearance based since we will develop in chapter.5, a novel approach which ignores such assumption. We describe below, the different training gaze samples available in public.

2.3 Gaze databases

For both approaches described previously, annotated gaze samples are needed to train the system and evaluate the accuracy of the gaze estimation. The majority of the methods based on the localization of facial feature points use specific databases to estimate robustly their locations. The appearance-based methods need training datasets with the gaze information as ground truth to build the mapping function. The representativity of these databases in terms of variability and quantity is directly involved in the efficiency of the system.

We describe some public gaze databases frequently used to train or evaluate automatic gaze estimation systems. For each database, we detail the acquisition setup used during the experiments.

2.3.1 Feature-based databases

- **Gi4E:** [Villanueva 2013] recorded 1339 images with a standard webcam corresponding to 103 different subjects with 13 images each. The images are manually annotated with eye feature points (pupil and corners).

  The images are provided in (800×600) resolution corresponding to different gazing points displayed on a 2D screen. The ground truth contains only the 2D locations of the eye facial points with no gaze information making this database available for eye-pupil detection learning algorithm and unsupervised appearance-based gaze methods. In Fig.2.8, we illustrate some samples from this database.

(a) (b)

Figure 2.8: Gi4E gaze database. (*a*) The annotated 2D eye feature locations for a given participants. (*b*) For each participant, different gazing configurations are recorded.

- **BioID:** consists of 1521 gray scale images of 23 different persons annotated manually with the 2D location of the eye-pupil and 18 2D facial points.

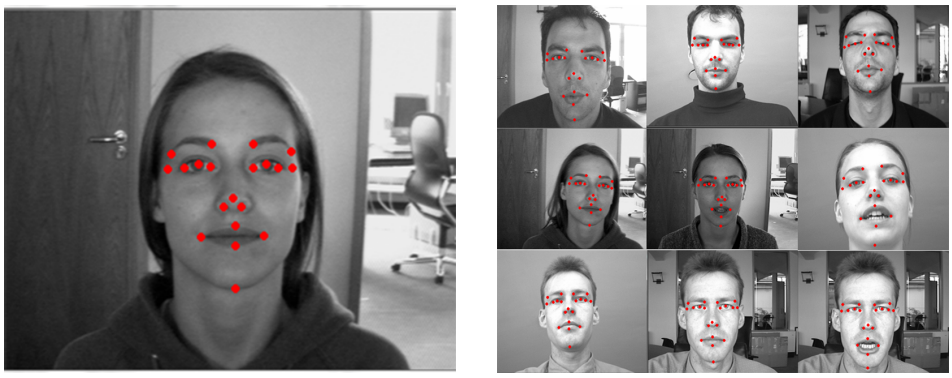
  The resolution of the images is (384×286) under important illumination, head pose changes and user-sensor distance variability. Initially used to compare the quality of face detection algorithms in unconstrained environment, it is used subsequently to evaluate the accuracy and robustness of eye-pupil localization algorithm, we give more details about the evaluation metrics performed in chapter.4. Fig.2.9 describes some extracted examples from this database with the facial annotated points, notice the apparent variabilities in the image.

- **MUCT:** to train a 2D flexible model able to capture the localization of the facial landmarks, [Milborrow 2010] recorded 3755 images with 76 facial points. The dabase contains a high diversity of lightning, age and ethnicity.

  The images are provided in (480×640) resolution corresponding to different head pose configuration. This database is usually used to train model to localize with a sufficient accuracy the eye points including pupils and corners. Fig.2.10 shows some sample with the corresponding landmarks.

- **Facial features-LFW:** Based on The Large database of Faces in the Wild designed by [Huang 2007] for studying the problem of unconstrained face recognition, [Dantone 2012] labels 13231 images of faces (including 1680 different persons) collected from the web with 10 facial feature points. This database is usually used to learn a robust mapping between the face appearance and the facial features location, yielding accurate eye points for feature-based gaze systems.

  The database presents a high appearance variability such as lightning condition, scale, head pose and presence of glasses. Fig.2.11 reports the nature of the existing variability in the annotated images.



(a) (b)

Figure 2.9: BioID database. (*a*) A total of 20 facial points are annotated including eye's points. (*b*) Each participants performs different conditions (user-sensor distance, head pose changes and illumination variation) to introduce variability in terms of appearance.

2.3.2 Appearance-based databases

- **UULM database:** [Weidenbacher 2007] recorded an extended dataset of 20 subjects including faces in various head pose configurations and eye gaze directions leading to a total amount of 2220 images. The images acquisition was under controlled conditions to adjust appropriately the subjects head and eyes. The images are manually labeled with landmarks indicating important features of the face.

  A digital camera with (1600×1200) resolution is used under three spotlights in a white canvas. To monitor the head pose, a laser pointer is mounted on the head of each subject displaying a red dot on the wall, leading to accurate head adjustment. For each head pose configuration, different gaze samples are performed (with two angular dimensions) using a laser water-level. For the head pose, 3 configurations are used for the pitch angle $\{-20°, 0°, +20°\}$ and 10 configurations for the yaw angle (from $0°$ to $90°$). 9 gaze samples are recorded for each head pose configuration. Fig.2.12 shows some examples extracted from this database.

- **Columbia database:** [Smith 2013] collected 5880 high-resolution images of 56 different people (32 male, 24 female). Different subjects appearance parameters are considered such as ethnicity (21 Asian, 19 White, 8 South Asian, 7 Black and 4 Hispanic), age (ranged from 18 to 36) and wearing glasses (21 with glasses).

  The acquisition was performed using a digital camera with a resolution of (5184×3456), the subjects used an adjustable chin rest to stabilize their faces. A $[7 \times 3]$ fixed grid of red dots is attached to a wall under a black background environment. The subjects were asked to fix the different red spots under

(a) (b)

Figure 2.10: MUCT database. (*a*) To build an Active Shape Model (ASM), [Milborrow 2010] used a et of 76 facial points. (*b*) Illustration of the annotated 2D eye feature locations .

different head pose configurations. For each subject, 5 gaze samples were recorded according to the 5 yaw values ($0°, \pm 15°, \pm 30°$). The experimental protocol and the final gaze sample images are shown in Fig.2.13.

- **EYEDIAP database:** Unlike the previous discussed databases, [Mora 2014] used an RGB-D camera (Microsoft Kinect sensor) coupled with an HD-camera to record gaze samples under different session scenarios. The database contains 16 different people (12 male, 4 female). A total of 94 sessions were recorded containing both RGB and depth video stream, HD video stream and ground truth files indicating the gaze information and the head pose parameters (estimated using a multiple instance fitting on a 3D morphable model strategy and ICP).

  The subjects were requested to gaze a predefined target (a 2D displayed point in the screen or a 3D tracked ball) with static and moving head pose activity under different illumination conditions and user-sensor distances. To synchronize the RGB-D (with 640×480 RGB resolution and 320×240 depth resolution) and the HD sensors, a set of 5 LEDs visible by both cameras was used. 2.14 illustrates the experiments setup and the obtained RGB-D samples.

- **MPIIGaze** [Zhang 2015] collected 213659 eye images of 15 participants during natural everyday laptop use over more than 3 months. Each participant performs between 1498 to 34745 samples. Each sample is annotated with 6 facial landmarks, and the corresponding 2D and 3D gaze point displayed on a screen (knowing the camera intrinsic parameters, a 3D position can be obtained).

  To obtain gaze samples, a specific software was implemented as a background service in the laptop of the user, asking to look at a random sequence of 20

(a) (b)

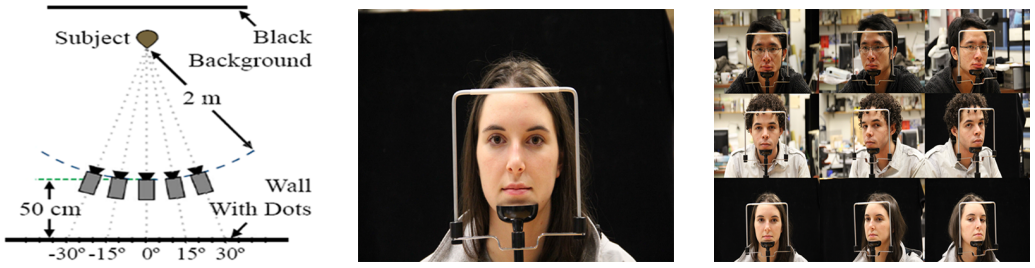
Figure 2.11: Facial features-LFW databse. (*a*) Each face is labeled with 10 facial landmarks including eye corners. (*b*) Annotated faces examples extracted from highly unconstrained environments.

(a) (b) (c)

Figure 2.12: UULM gaze and head pose database. (*a*) Overview of the setup used for the experiments. (*b*) Each image is labeled with 9 facial landmarks. (*c*) Some example of the final result images.

positions. To ensure that participant are concentrated on the task, they confirm by pressing the spacebar. This dataset presents more variability in terms of appearance compared to the previous databases. The main motivation of the authors was to provide a significantly important training set to train a convolutional neural networks as a mapping function between the different eye appearances and gaze information.

Despite the efforts provided in building the previous discussed databases, some challenging problems in training gaze estimation systems are still unsolved. Indeed, providing sufficiently representative training gaze samples in terms of quality (accurate ground truth and high appearance variabilities) and quantity is a very tedious task taking an important time to achieve reliable data.



(a) (b) (c)

Figure 2.13: c. (a) Illustration of the setup used for the experiments. (b) An adjustable chin rest is used to stabilize the head. (c) Some example of the final gaze sample images.

(a) (b) (c)

Figure 2.14: EYEDIAP gaze database. (a)The setup used for the experiments. (b) Multiple instance fitting to get a specific 3D head pose model to estimate the head pose parameters in 3D eyeball gazing session. (c) shows some of the final result images with RGB-D information.

2.4 Conclusion

The automatic gaze estimation systems have recently experienced an important progression in terms of algorithm robustness and hardware acceptability. Nevertheless both feature-based and appearance-based approaches are still presenting some limitations under unconstrained environment. For the first category, the accuracy is strongly constrained by the facial points extraction and localization. Despite the recent promising results obtained in this field such as in [Dantone 2012], [Xiong 2013], localizing such features under strong head pose variations is still a very challenging task. To get around this problem, the second family formalizes the gaze estimation as a learning problem achieving slightly more accuracy in head pose scenarios.

Appearance-based methods solved the problem of feature localization by formulating the problem differently. One of the most challenging problem of such approach lies in providing reliable data. Indeed, to achieve a robust mapping function with high ability of generalization, the training data have to satisfy sufficient variability in terms of appearance and high cardinality. Moreover, these methods

(a) (b) (c)

Figure 2.15: MPIIGaze database. (*a*) a designed software asks the participant to look at a specific position on the laptop screen. (*b*) in addition to the 2D and 3D gaze estimation, 6 facial points are given (estimated automatically using the method from [Baltrušaitis 2014]). (*c*) illustrates participant with high appearance variability (right column) and the extracted eye image provided in the database used for the learning.

as the feature-based, usually assume head pose and eye appearance as independent blocks. Concretely, they learn the mapping in frontal configuration then perform a geometric correction based on head pose parameters. Such paradigm produces a cumulated error which is a direct consequence of cascading the two components. In addition the accuracy of these methods decreases significantly in large user-sensor distance scenarios.

In our case, as we want to deal with highly unconstrained environments, we decide to revisit the classical approaches. We target in each approach the important components in the processing chain and propose ameliorations. Then, we propose a novel paradigm that can overcome the limitations of the traditional designing of automatic gaze estimation systems. We make the assumption of a global gaze manifold with head and eyes cues unified in a single block. To capture the distribution of such manifold, existing gaze databases do not meet our learning requirements. Thus, we decided to use computer graphic rendering techniques to generate a sufficient amount of labeled data to obtain sufficiently robust gaze estimation. We also build our own real RGB-D gaze database which follows a rigorous protocol to obtain sufficiently reliable gaze samples data.

Chapter 3

Random Forest algorithm

Contents

All the investigations and experiments achieved in this work are strongly based on Random Forest algorithm. In this chapter we describe in details this machine learning tool. In Sec. 3.1 we give a comprehensive overview and provide the mathematical notations which formalize analytically Random Forest tasks in Sec. 3.2. Sec. 3.3 compares Random Forest discriminative performance to some alternative algorithms. Sec. 3.4 reports some recent improvements on this algorithm especially in computer vision and medical image analysis fields. We conclude this chapter in Sec. 3.5.

3.1 Overview

Learning decision trees from data has been a long standing problem. One of the pioneering works trying to solve such problem is Classification And Regression Trees (CART) in [Olshen 1984] describing the basics of decision trees and their use in classification and regression tasks. Several ameliorations were proposed in order to enhance the accuracy of the decision such as C4.5 presented in [Quinlan 1993] which are considered as the most popular algorithms in this field. To make a final decision, the trees are used as individual predictors.

One of the important limitation of this earlier work in decision trees is the ability of generalization, which measures the accuracy of the prediction across unseen scenarios (testing data which are not used during the learning step). Recently, the use of trees as an ensemble of predictors has emerged producing more robust and accurate final prediction. In this case, each tree is considered as a weak learner, a typical method using this idea is the boosting algorithm presented in [Schapire 1990] which performs an iterative re-weighting on training data to learn trees as weak predictors, a linear combinations of these trees is performed to build a strong classifier. In computer vision, one of the most poplar work following such strategy is the Viola and Jones face detector described in [Viola 2001].

Breiman introduces Random Forest model in his work [Breiman 2001] which is an ensemble on randomly trained trees. The author introduces two levels to inject randomness during the learning of the trees which are then grouped in a single forest. First via randomizing feature selection, second by randomly sampling the labeled training data. Merging all the trees decision yields superior generalization compared to the other decision trees models. To measure the robustness and correlation between predictors, the author describes some cost functions based on prediction error for this purpose.

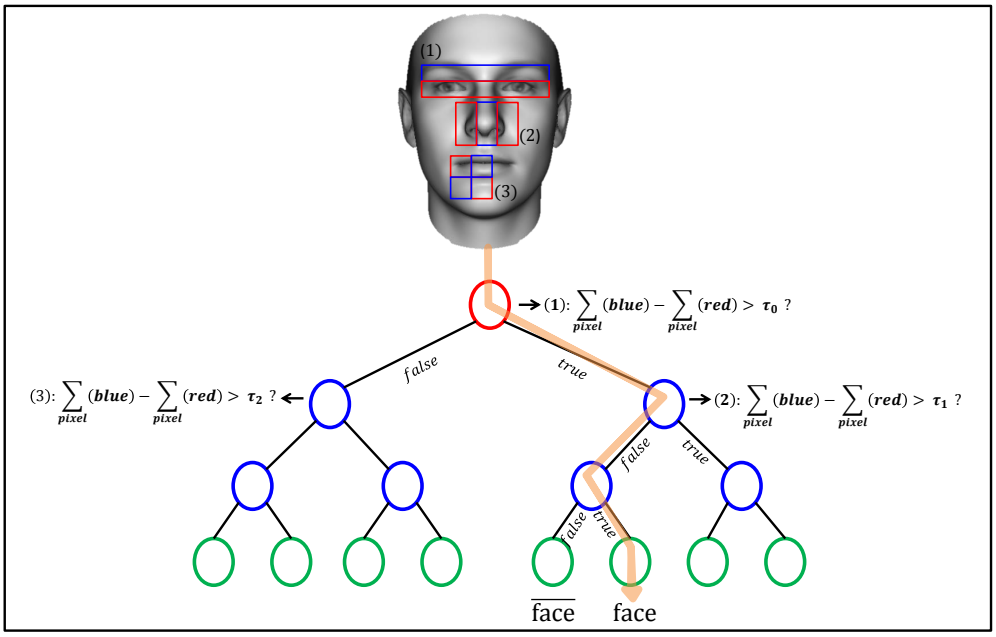
These last years, using Random Forest decision in machine learning, computer vision and medical image analysis have seen an explosive growth. Among the most successful applications of the Random Forest algorithm, we notice the human body part classification using Kinect sensor in [Shotton 2013].

3.2 Mathematical model

We describe here Random Forest as a mathematical tool, we present a unified notation which allows us to formalize clearly some machine learning tasks such as classification, regression, density estimation or manifold learning.

3.2.1 Decision tree basics

A decision tree is a technique which splits the initial problem into two low complex problems in a recursive way. At each node, a simple binary test is performed. According to the result of the test, a data sample is directed towards the left or the right child. The tests are selected to achieve an optimal clustering. The terminal



(a)

Figure 3.1: The hierarchical structure of a binary tree. The nodes are represented by circles, the root node in red, internal nodes (split nodes) in blue and terminal nodes (leaves) in green. An illustrative example of a decision tree, each split node stores (at the training step) and performs (at the testing step) a binary test to the incoming data, the leaves save the prediction model. In this example, we show a simple way of using a decision tree for a face detection problem.

nodes of the tree, called leaves, store the estimation models approximating the best desired output. In Fig. 3.1 we formalize a face detection problem using a decision tree. At the testing step, a difference of integral images (between red and blue boxes) extracted from an image test is compared to a random threshold value τ at each node. Each node is associated to a unique binary test. According to the results of these tests, the data is directed following a path toward a given leaf (represented in an orange arrow) which gives the prediction answer, namely classifying the image as face or not. The binary tests and the prediction parameters are fixed during the training phase in an optimal way which maximizes the data clustering.

3.2.2 Random Forest notation

A predefined notation is primordial to formalize clearly a given problem tackled with Random Forest algorithm.

We denote a generic input data point by a vector $\mathbf{x} = (x_0, x_1, ..., x_d) \in \mathbb{R}^d$ where d represents the dimensionality of $\mathbf{x}$, also called feature space dimensionality. $\mathbf{y}$ represents the output data which describes the associated information related to $\mathbf{x}$ (also called the label, it can be discrete, continuous or not provided depending on

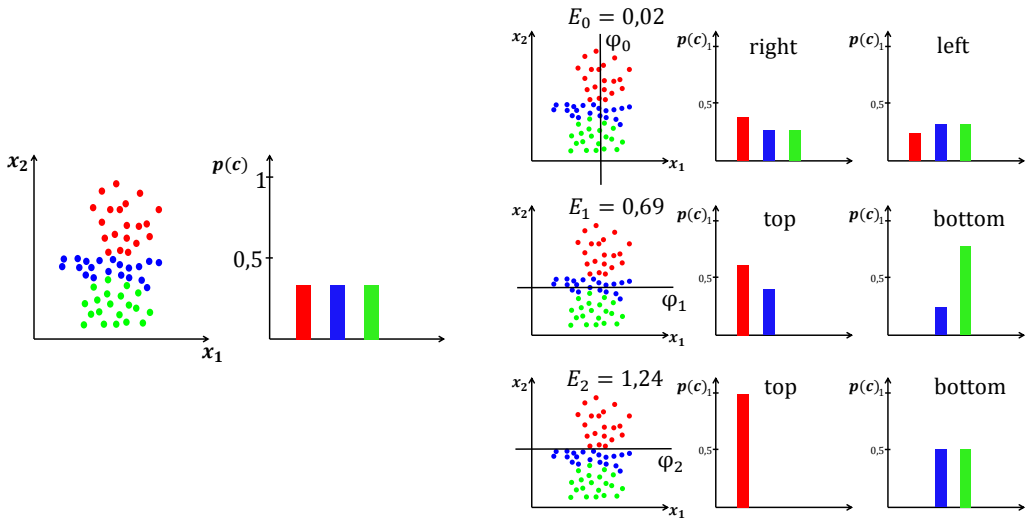


Figure 3.2: Information gain for discrete distribution. For each binary test (splitting) ϕ_i, a correspondent information gain E_i is calculated. The optimal splitting corresponds to the highest value.

the handled task)

Training is an off-line step which fixes as mentioned previously, the binary tests at the internal nodes and the prediction decision at the leaves.

Given a training set $\mathcal{P}_j$ of data points $\{(\mathbf{x}, \mathbf{y})\}$ at a given node j, a binary test $h(x, \phi) \in \{0, 1\}$ is applied at the node j to split data into two subsets $\mathcal{P}_j^{\mathcal{L}}$ and $\mathcal{P}_j^{\mathcal{R}}$ going to the left and right children of that node respectively. $\phi = (\psi, \tau)$ defines the splitting parameters with ψ describing a feature selection function and τ is a random scalar value. The ensemble of possible ϕ cardinality is defined by ρ which controls the level of randomness allowed to the trees.

The fact that the trees are binary implies the following properties $\mathcal{S}_j = \mathcal{P}_j^{\mathcal{L}} \cup \mathcal{P}_j^{\mathcal{R}}$ and $\mathcal{P}_j^{\mathcal{L}} \cap \mathcal{P}_j^{\mathcal{R}} = \emptyset$. To optimize the node parameters by finding the optimal test ϕ^*, an energy function E is used as a training objective function. The information gain is widely used in decision forest optimization as energy function defined as follows:

$$\phi^* = argmax(E_j) \tag{3.1}$$

with

$$E_j = H(\mathcal{P}_j) - \sum_{i \in \{\mathcal{L}, \mathcal{R}\}} \frac{|\mathcal{P}_j^i|}{|\mathcal{P}_j|} H(\mathcal{P}_j^i) \tag{3.2}$$

where $H(\mathcal{P}_j)$ defines the entropy which depends on the tackled task.

In Fig.3.2, we illustrate, for a classification problem where the entropy is defined as the Shanon entropy $-\sum_c p(c) \log_2(p(c))$, the existing correlation between the splitting discrimination and information gain. A good separation of the data

reflected by clustering a minimum of classes in each node gives the highest information gain. This characteristic is also known as the node impurity. All the tree nodes are optimized in the same way sequentially. To stop growing a tree, many criteria can be used such as maximum depth tree $\mathcal{D}$ which is a predefined value, or when the number of training data $|\mathcal{P}|$ let down a predefined threshold.

Testing is a runtime step, once all the nodes parameters for each tree in the forest are optimized, a decision forest can be applied. Given an unseen testing point **x'**, each tree applies at each node the fixed split function to **x'** starting from the root. This operation is repeated until reaching a terminal node which contains a prediction model. All the returned leaves of the trees are averaged for a global decision. For instance, in a discrete distribution as discussed in Fig.3.2, each tree χ in the forest casts a prediction $p_\chi(c|\mathbf{x})$. The final decision can be expressed as follows:

$$p(c|\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} p_\chi(c|\mathbf{x}) \tag{3.3}$$

where T is the number of the trees in the forest. The fact that the trees are trained separately verifying statistical independence, an other way to average the decision from all the trees can be expressed as follows:

$$p(c|\mathbf{x}) = \frac{1}{K}\prod_{t=1}^{T} p_\chi(c|\mathbf{x}) \tag{3.4}$$

K is a probabilistic normalization constant.

3.2.3 Random Forest tasks

As the global notation is defined, Random Forest can be considered as a flexible model able to handle different problems according to the nature of the input data. Fig.3.3 summarizes some problems that can be tackled successfully by Random Forest.

Classification A classification forest aims to learn a mapping which associates unseen test data with their correct classes. The mapping is learned from a labeled training set $\mathcal{P} = \{(\mathbf{x}, c)\}$ where c represents a discrete value (instead of using $\mathbf{y}$, c describes the class of $\mathbf{x}$). The information gain defined in Eq.3.2 is calculated using the Shannon entropy. The mapping between an unseen point and its class is expressed in a probabilistic way through the Eq.3.3 or Eq.3.4. Fig.3.3 illustrates a Random Forest classification problem with 2D data organized in 03 classes. Some relevant works in computer vision using classification forest are presented in [Lepetit 2005] and [Marée 2007].

Regression A regression forest aims to learn a mapping which associates unseen test data with their correct continuous prediction. The mapping is learned from a labeled training set $\mathcal{P} = \{(\mathbf{x}, \mathbf{y})\}$ where $\mathbf{y}$ represents a continuous variable in $\mathbb{R}^n$ describing the label of $\mathbf{x}$. $\mathbf{y}$ is usually assigned to a random continuous variable

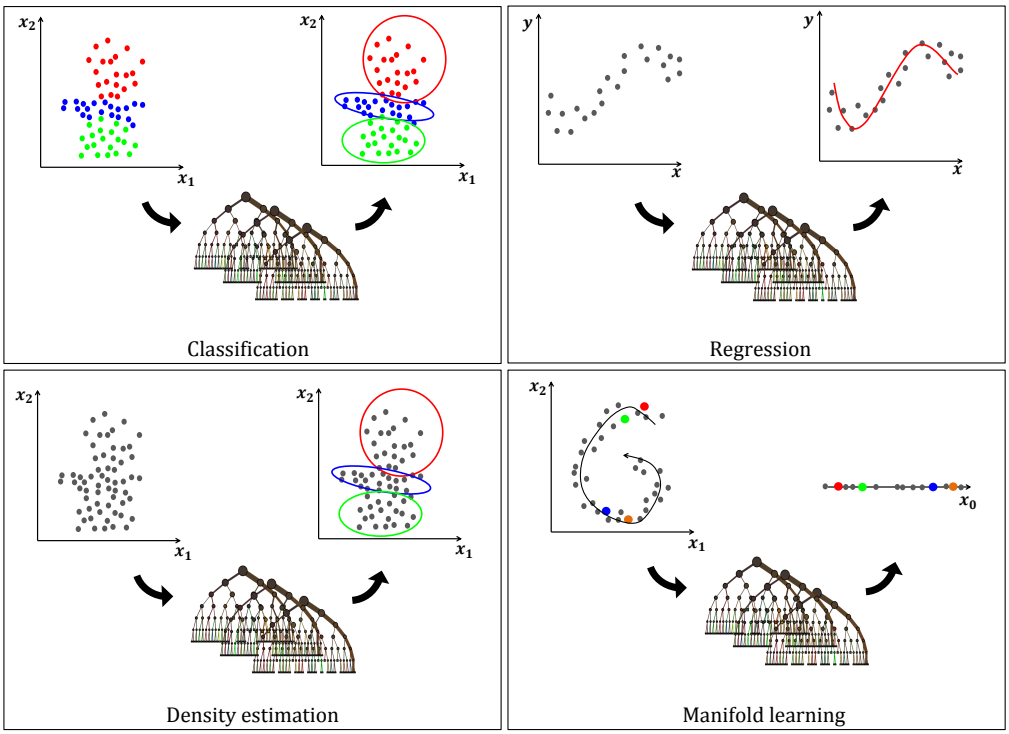


Figure 3.3: Random Forest framework presents a high flexibility, different tasks can be tackled depending on the nature of the input data.

which allows to re-write the information gain for the nodes optimization as follows:

$$E_j = \log(|\Sigma_y(\mathcal{P}_j)|) - \sum_{i\in\{\mathcal{L},\mathcal{R}\}} \frac{|\mathcal{P}_j^i|}{|\mathcal{P}_j|} \log(|\Sigma_y(\mathcal{P}_j^i)|) \tag{3.5}$$

where $\Sigma_y(\mathcal{P}_j)$ is the covariance matrix related to y at the node j and $|.|$ is the determinant operator. Another error based on euclidean distance can be used as follows:

$$E_j = \sum_{\mathbf{y}\in\mathcal{P}_j} (y-\bar{y}_j)^2 - \sum_{i\in\{\mathcal{L},\mathcal{R}\}} \left(\sum_{\mathbf{y}\in\mathcal{P}_j^i} (y-\bar{y}_j)^2 \right) \tag{3.6}$$

where $\bar{y}_j$ indicates the mean of the random variable y at the node j. To perform a regression prediction for an unseen point $\mathbf{x}$, the leaves store mostly a multi-variate Gaussian distribution of y, then each tree prediction $p_\chi(\mathbf{y}|\mathbf{x})$ can be assigned to a distribution $\mathcal{N}(y, \bar{y}, \Sigma_y)$. Then, the final estimation is given by averaging all the trees following Eq.3.3. Fig.3.3 gives an overview of random forest-based 1D regression. Some popular works achieving good regression generalization [Fanelli 2011] and [Shotton 2013].

Density estimation A density forest aims to estimate a density function from which unlabeled data have been generated. From the unlabeled training set $\mathcal{P} = \{\mathbf{x}\}$, the density function $p(\mathbf{x})$, in contrast with tree prediction, is learned. To

optimize node parameters for each tree, the same objective function as in regression is used. Knowing that the ground truth labels are not provided, the determinant of the covariance matrix $|\Sigma(\mathcal{S}_j)|$ corresponds to the volume of the ellipsoid generated by data at the node j. A final clustering is given by averaging the density of each tree predictor. In Fig.3.3, we show an example of density forest with the same set of training data as in classification but with no ground truth labels, as done in [Moosmann 2007] and [Ram 2011].

Manifold learning A manifold learning forest aims to project the input data $\mathbf{x} = (x_0, x_1, ..., x_d)$ into a novel space represented by $\mathbf{x}' = (x'_0, x'_1, ..., x'_{d'})$ such that $d' << d$ with preserving the relative geodesic distances in the initial space. From a set of training data $\mathcal{P} = \{\mathbf{x}\}$, a novel representation can be learned. The same objective function as in the density estimation forest is used to optimize the nodes, to preserve the geodesic distances, a measure of similarity between the initial points and their projections is performed. Each tree χ captures the similarity of the reached data as a $k \times k$ affinity matrix W^χ (k defines the cardinal of the set of training data $|\mathcal{P}|$) where $W^\chi_{ij} = e^{Q^\chi(\mathbf{x}_i, \mathbf{x}_j)}$. Q is a metric distance, [Criminisi 2011] defines different ways of calculation, Mahalanobis, Gaussian and binary. Fig.3.3 gives an illustrative example of a manifold learning forest projecting input data from 2D space into 1D. An application of this forest can be found in [Gray 2013].

3.3 Random Forest versus alternative algorithms

To compare decision forest to different machine learning algorithms, we follow the same approach as in [Criminisi 2011]. We reproduce their illustrative results using similar training data. We first use the Microsoft sherwood[1] library to train our model, we use in a second time, our own implementation to validate the result for each comparison.

3.3.1 Support Vector Machine

We perform the comparison using two training data sets. Each training set is a four-class 2D point cloud easily separable. We train the SVM model[2] in a one-vs-all configuration. According to the results reported in Fig.3.4, both techniques produce good separation for both experiments. Based on these separations, one relevant difference can be established, Random Forest model produces an important additional information which is uncertainty. This characteristic is naturally linked to the appearance of the training samples, uncertainty increases as moving away from the data. Unlike the separation learned by the forest, SVM model assigns an equal confidence and produces a hard boundary between the classes. [Gall 2013] compares Random Forest and SVM accuracy under action recognition task, and [Shotton 2013] performs a comparison under a human 3D body part classification.

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52340

[2] We use the publicly available code http://asi.insa-rouen.fr/enseignants/ arakoto/toolbox/

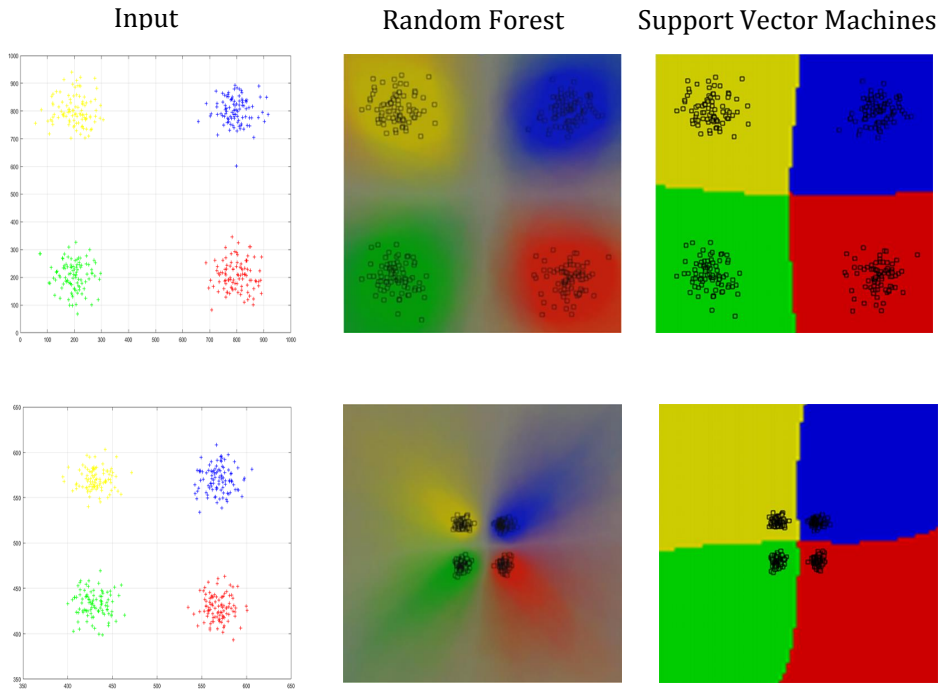


Figure 3.4: 4-classes classification problem using Random Forest and SVM respectively. Random Forest produces smoother separation between classes.

3.3.2 Gaussian Process

To illustrate the ability of the regression forest to regress continuous output, we perform two experiments under different training datasets and compare to Gaussian process model. To perform the GP learning, we use the publicly available library Gaussian Process for Machine Learning (GPML[3]) with optimal parameters.

According to Fig.3.5, both Random Forest and Gaussian Process capture the global regression pattern of the points distribution (represented by the green curve). The two models capture the uncertainty as moving from the training data (represented by brown regions). For the Random Forest, gray curves represent the estimated mode which demonstrates the ability to capture multi-modal distributions (bi-modal distribution in the example) while GP produces uni-modal predictions. These results can be explained by the piece-wise nature of the trees which is more apt to model behaviors with different modalities. One of the intrinsic characteristics of the GP is the uni-modality which can be a real limitation in ambiguous data configurations.

[3] http://www.gaussianprocess.org/gpml/code/matlab/doc/

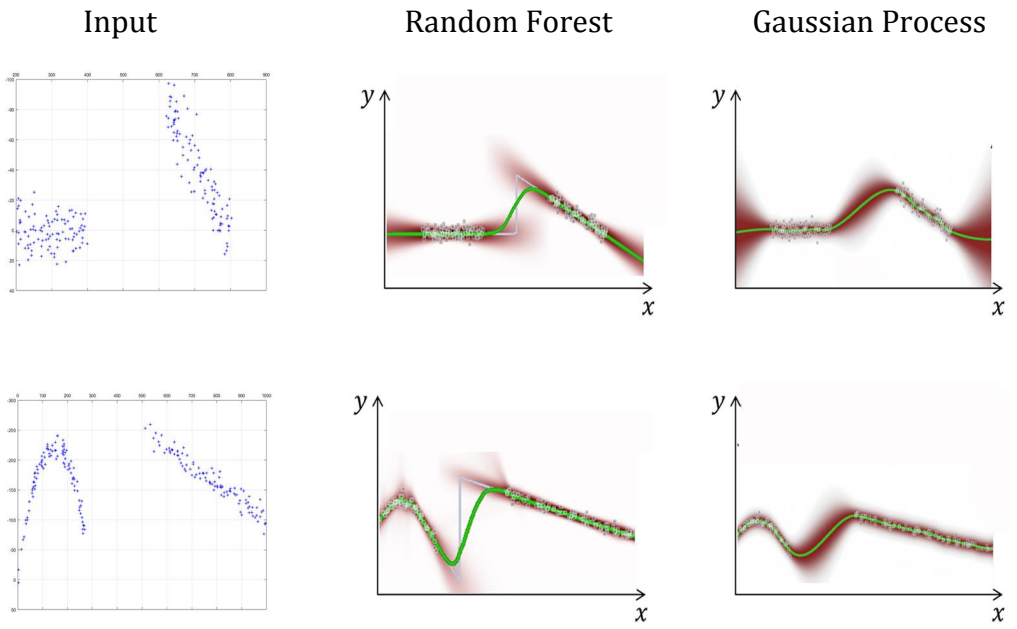


Figure 3.5: 1-D regression using Random Forest and Gaussian Process under two two training data sets. Both models produce uncertainty around regions with less training data, however Random Forest captures the multi-modality of the data distribution when GP is highly constrained by uni-modal predictions.

3.3.3 K-Nearest Neighbour

Fig.3.6 shows a comparison between Random Forest and k-Nearest Neighbour (KNN) density estimation under two training sets. To perform the KNN clustering, we use Statistics Matlab Toolbox[4]. The first experiment points are sampled from five-Gaussian distributions, the second point cloud is repartitioned along four spiral distributions. The figure illustrates the ability of the forest to build smooth outputs that capture nicely the input points for both experiments. The KNN estimator produces some artifacts and deformations in the final output, as finding the optimal value of k is very challenging, which directly affects the prediction. The smoothness ability of forest is the result of the involvement of several decorrelated tree predictors. Using a highly optimized single tree will produce similar problems as in KNN.

3.3.4 Gaussian Mixture Model

We compare Random Forest density estimation to Mixture Gaussian Model (GMM) using the same two training sets as in the previous comparison, and under a third training set arranged in 'S' shape as described in Fig.3.7. We trained our GMM

[4]http://fr.mathworks.com/help/stats/kmeans.html

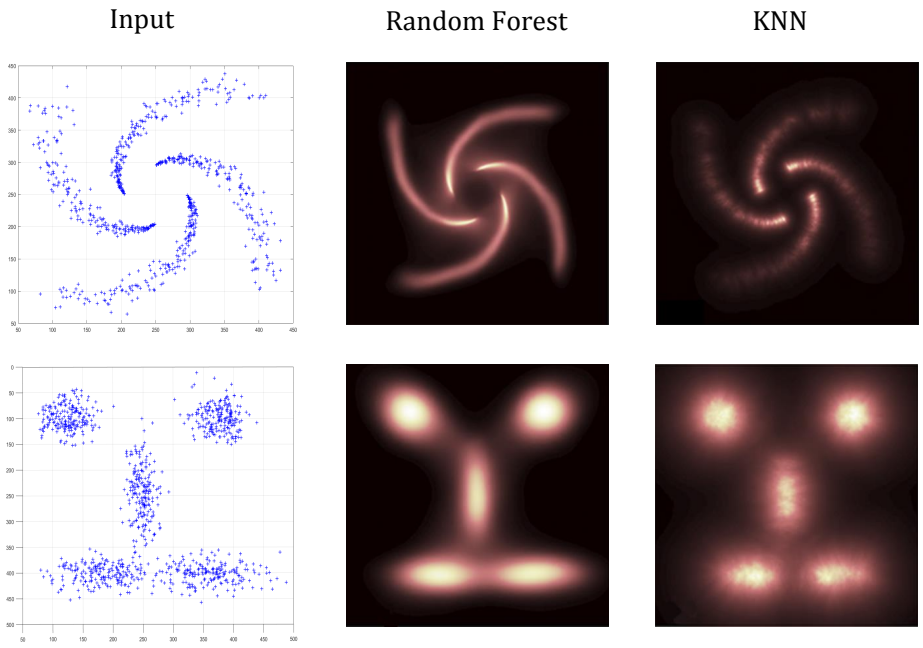


Figure 3.6: Density estimation comparison between Random Forest and non-parametric estimator. Random Forest parameters are fixed to 200 trees with a maximum depth of 5. For k-Nearest Neighbour, we fix k to 15 and 60 respectively (these values seem to achieve the best result). For both distributions, Random Forest prediction is more smoother than KNN.

models using the same Matlab toolbox as in the previous comparison with an Expectation-Maximization (EM) optimization. According to the reported results in Fig.3.7, both Random Forest and GMM capture nicely the distribution of points sampled from the two last training sets (the smoothness is globally similar). However, for the first configuration, the GMM prediction produces slight artifacts compared to the Random Forest output. This GMM behavior under this configuration is probably linked to the EM convergence. Indeed, the EM optimization is very constrained by the Gaussian parameters initialization, the convergence can fail in local minima.

[Criminisi 2009] reported a Random Forest and GMM comparison under an automatic 3D localization of human organs using CT volumes as input.

3.4 Advanced Random Forest

Recently, Random Forest experienced many improvements especially in computer vision and medical image analysis according to the growing challenges. We describe here some relevant contributions allowing to enhance the potential of this technique and to overcome some limitations.

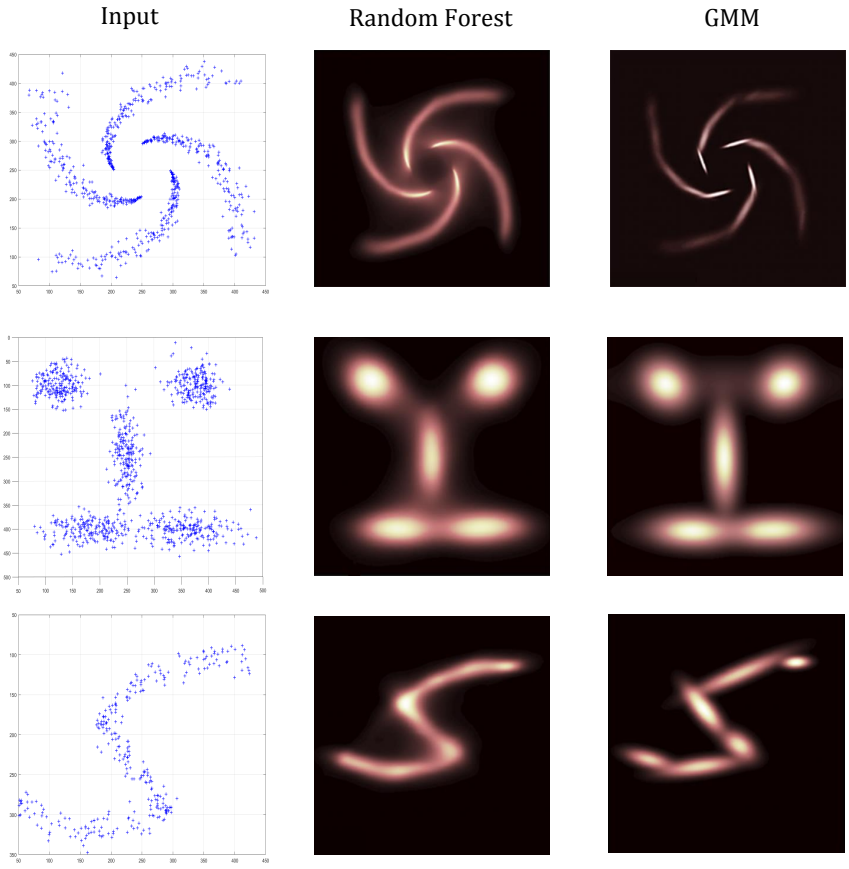


Figure 3.7: Density estimation comparison between Random Forest and Gaussian Mixture Model. We use the same parameters as the previous comparison for the forest. For the GMM, we fix the number of Gaussian to 17, 7 and 6 across the different experiments respectively. The two models capture nicely the different distributions with a good amount of smoothness. Nevertheless, GMM produces some artifacts in some configurations as in the first experiment compared to Random Forest.

3.4.1 Extremely Randomized Forest

Introduced by [Geurts 2006], Extremely Randomized Forest is an ensemble of highly decorrelated predictors trained under a very weak node parameters optimization. This decision forest can be considered as particular instance of the initial Random Forest algorithm. To illustrate the amount of randomness introduced for each tree at each node, we use an additional parameter ρ as done in [Criminisi 2011] and defined previously. ρ describes the entire ensemble of the possible values of ϕ. In standard Random Forest decisions, ρ takes a relatively high value ($100 \sim 10k$) which presents a reasonable randomness. Such characteristic produces a high probability of sharing similar behavior across different nodes and different trees (*i.e.*, by sharing similar features and thresholds at the splitting). In the case of extremely randomized

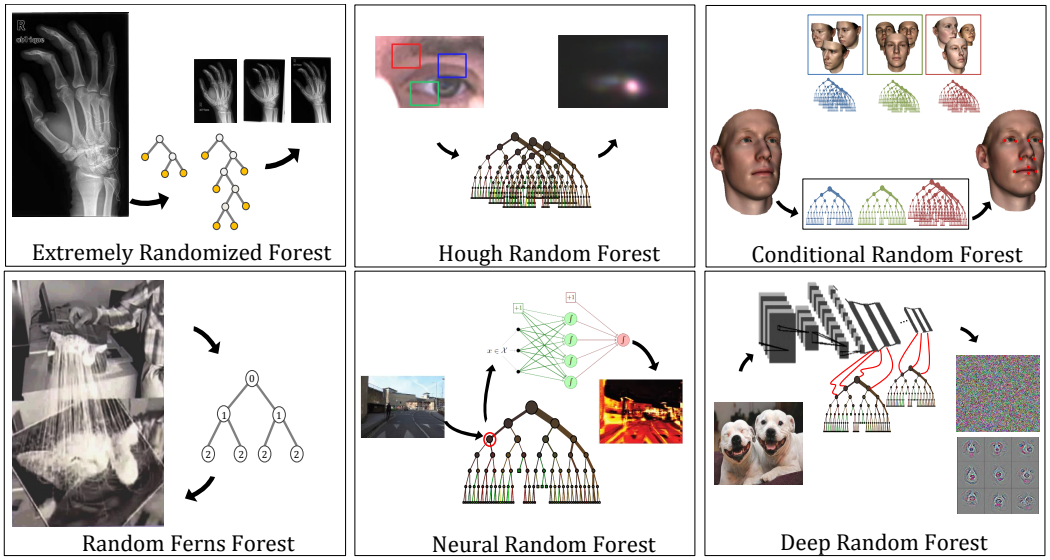


Figure 3.8: Some important contributions on Random Forest algorithm this last decade. Extremely randomized trees is a specific instantiation of Random Forest with a maximal randomness in node optimization process ([Marée 2007] applied it for medical image retrieval). Random ferns is another instantiation constrained by the assumption of performing the same node optimization parameters at each tree level ([Ozuysal 2010] applied it for image classification task). Hough forest is a combination of standard trees and Hough voting strategy, each tree saves regression and classification information ([Kacete 2016] applied it for human eye pupil localization). Conditional forest decision, trees are selected using prior knowledge to infer the final estimation ([Dantone 2012] used head pose parameters as prior to estimate facial points robustly). In neuronal forest, a novel representation of data is learned in the splitting nodes using Multi Layer Perceptron ([Bulo 2014] applied it to semantic image labelling). [Kontschieder 2015] used a deep approach based on convolutional neural network to learn the novel representation.

trees, randomness is maximal ($\rho \rightarrow 1$). Moreover, the trees are learned on the entire training set with no bootstrapping. Fig.3.9 gives an example of extremely randomized trees, notice the apparent difference between the predictor in terms of maximum depth and number of leaves (represented as yellow circles).

Such approach can be found in [Marée 2007] for image classification and retrieval task.

3.4.2 Random Ferns Forest

To recognize and localize some key points in images, [Ozuysal 2010] learned specific trees called ferns. The fundamental assumption of this approach lies in the fact that nodes belonging to the same level in the tree are forced to share the same splitting

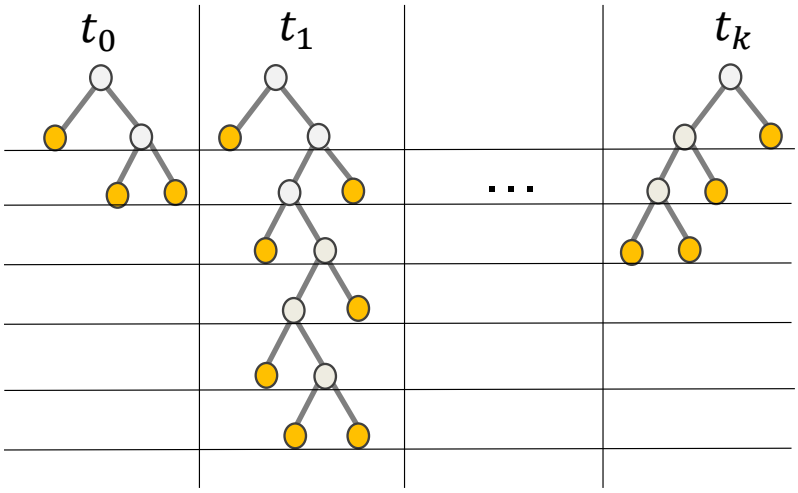


Figure 3.9: An example of an ensemble of extremely randomized trees. Increasing the amount of randomness at each node optimization produces highly decorrelated trees. The difference lies in maximum depth reached or in the number of leaves.

parameters ϕ. One of the advantages of this modification of the initial algorithm is reducing the training step computational time without losing any discriminative performance. Fig.3.10 establishes the difference between a standard tree and a fern under classification problem. To separate the different classes, a standard tree performs less processing in terms of levels to capture the classes boundaries compared to the fern. As a direct consequence of sharing ϕ across the different nodes, the splitting form is necessarily a complete hyperplane. According to this behavior, ferns require more depth to capture the different classes configurations as performed by the standard trees. Facing the problem of lack of training data, ferns suffer less from overfitting problems than Random Forest.

[Pauly 2011] performed the same approach under human organ localization task formulated a regression problem.

3.4.3 Hough Random Forest

Introduced by [Gall 2013], Hough Random Forest is a combination of a classical decision forest discussed previously and Hough-transform voting strategy. The principal motivation of this method is to detect object parts with a very discriminative codebook. To build such codebook, the authors trained trees able to capture both classification and regression information. Instead of making decision based on the whole image as input, an ensemble of patches is extracted and processed separately by the trees. At training step, the trees are built based on a collection of patches, the leaves store a prediction model allowing to cast a probabilistic vote about the class and the localization of the object with small uncertainty. A Hough image, as

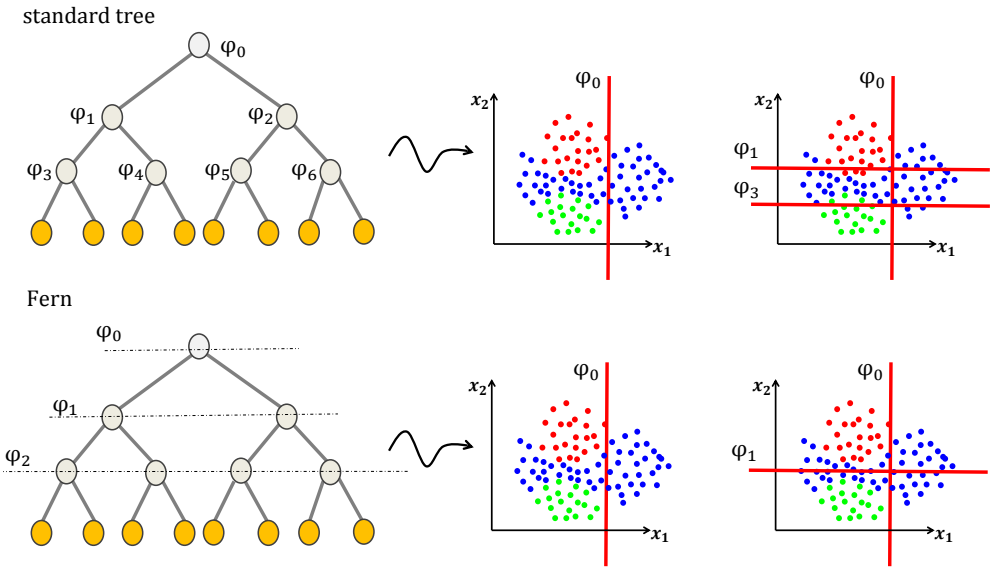


Figure 3.10: Random ferns paradigm. Unlike standard tree, the splitting nodes of the fern share the same optimization parameters ϕ. For a classification problem, trees capture different classes boundaries where ferns require more depth to achieve best results.

a voting space is built with accumulating all votes related to the patches. To find the maxima in this space which correspond to the object location, a non-parametric clustering is usually used. The authors used a Parzen-windowing with a Gaussian kernel, each tree prediction can be re-written as follows:

$$p(\mathbf{y}|\mathbf{x}) = \left(\frac{1}{|D_L|} \sum_{m \in D_L} \frac{1}{2\pi\sigma^2} \exp\left(- \frac{||(\mathbf{y}_0 - \mathbf{y}) - m||^2}{2\sigma^2} \right) \right) . C_L \tag{3.7}$$

where σ represents the covariance matrix of the Gaussian kernel. D_L is a set containing all the offset distances between the patches centers reaching a leaf during the training step (with m as a single offset). $\mathbf{y}_0$ is the center of the patch $\mathbf{x}$. C_L is a classification quantity measuring the proportion of the object class.

3.4.4 Conditional Random Forest

[Dantone 2012] and [Sun 2012] injected a conditional model to the regression forest decision to handle facial landmarks and human body parts detection. The main idea of this method is about making the assumption that the leaves outputs are dependent through a global latent variable, unlike previous random forest methods which work under the strong assumption that the outputs are independent variables. The authors trained trees able to incorporate prior knowledge about the tackled problem (for instance head pose parameters for the first work and body measurements

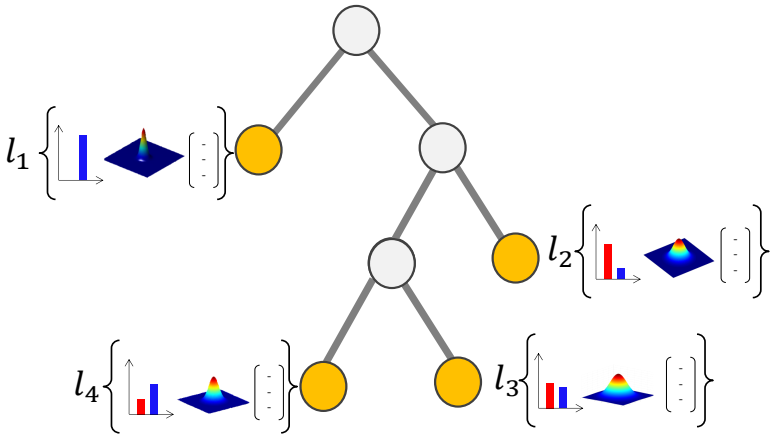


Figure 3.11: Example of Hough tree, each leaf contains classification (the classes probabilities), regression (multivariate Gaussian distribution) and offsets (set of distances between patches centers and object centroid) information.

for the second one) in order to build a conditioned relation between each leaf node decision and the latent global. In others words, the trees do not have to deal with all variability (facial or body variability respectively) which yields a more robust and accurate decision. $p_\chi(\mathbf{y}|\mathbf{x})$ can be formulated as follows:

$$p_\chi(\mathbf{y}|\mathbf{x}) = \int p_\chi(\mathbf{y}|\omega, \mathbf{x}) p_\chi(\omega|\mathbf{x}) d\omega \tag{3.8}$$

where ω is an auxiliary variable which encodes prior knowledge in practice. ω corresponds to the head pose in [Dantone 2012] and body measurement such as height in [Sun 2012]. $p_\chi(\mathbf{y}|\omega, \mathbf{x})$ is learned in both works while $p_\chi(\omega|\mathbf{x})$ is learned in the first work and given in the second one.

3.4.5 Neural Random Forest

[Bulo 2014] presented a novel approach using Random Forest with Multi Layer Perceptrons (MLP) for a semantic image labeling. Unlike previous methods where the input space (data representation) is left unchanged throughout training and testing step, the authors proposed to use the MLP as a split function within the internal nodes. These split nodes learn possible novel non-linear representations of the data for a better optimization of the nodes parameters and a more discriminative prediction at the leaves level. Unlike standard decision forest which considers a deterministic splitting at each node, neural tree performs a stochastic separation f_i on the data reaching each node. f_i represents the non linear activation response to the final representation of x defined as follows:

$$f_i = \sigma(\theta_i^\top x) \tag{3.9}$$

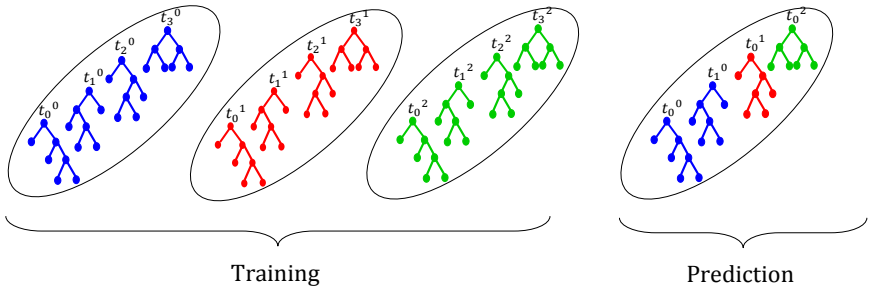


Figure 3.12: An illustration of a conditional decision with an ensemble of trees. For each configuration j, which relates to an auxiliary variable, an ensemble of specific trees $\cup t_i^j$ is trained. At the testing time, an information according to the auxiliary variable is injected in the global prediction selecting the right proportion of trees for each configuration.

where:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.10}$$

This separation yields a data distribution $\pi = \{\pi^{\mathcal{R}}, \pi^{\mathcal{L}}\}$ related to right and left child respectively. Fig.3.13 illustrates an example of a neural trees with a stochastic splitting. To find the best split choice, the likelihood $Q(\Theta)$ corresponding to a possible choice of Θ and π is performed as a loss function:

$$Q(\Theta) = \max_{\pi} \mathbb{P}(y|x, \pi, \Theta) \tag{3.11}$$

where $\mathbb{P}(y|x, \pi, \Theta)$ encodes a single tree decision which can be expressed as follows:

$$\mathbb{P}(y|x, \pi, \Theta) = \sum_{d \in \{\mathcal{R}, \mathcal{L}\}} \pi^{(d)} f_d(x, \Theta). \tag{3.12}$$

The MLP parameters Θ and distributions π are optimized alternatively using a regularized Back-Propagation and a concave maximization (based on a multiplicative update) respectively.

3.4.6 Deep Random Forest

Introduced by [Kontschieder 2015], Deep decision forests is a novel approach that unifies the ability of discrimination of the standard classification trees and the learning of novel data representation ability of the deep convolutional networks. One major difference compared to the neural decision forest lies in the learning strategy. Instead of optimizing Θ and π (the parameters related to the network and

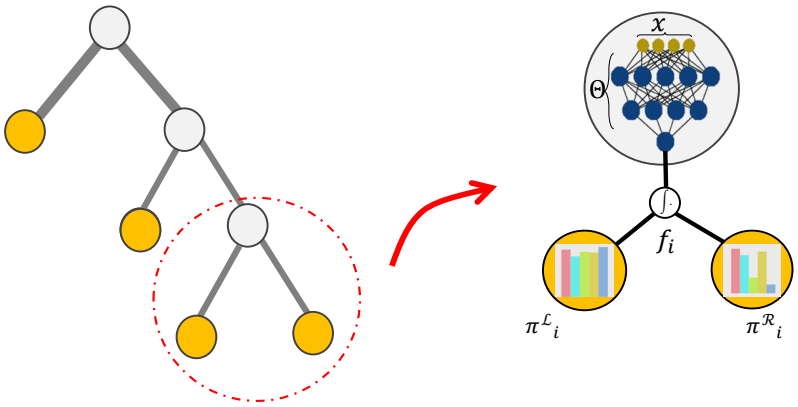


Figure 3.13: Visual illustration of a neural tree. Each binary splitting node is performed by a Multi Layer Perceptron described by the parameters Θ. The MLP learns a possible novel data presentation which allows more discriminative separation (described here by the two distribution $\pi^{\mathcal{R}}$ and $\pi^{\mathcal{L}}$)

trees respectively) alternatively, the training is performed in an end-to-end manner. Fig.3.14 describes the overview of this approach. The final outputs d_i of the network are directly linked to forest splitting nodes after performing a sigmoid operation with: .

$$d_i(x, \Theta) = \sigma(f_i(x, \Theta)) \tag{3.13}$$

where $\sigma = (1 + e^{-x})^{-1}$.

An other difference compared to the neural decision forest is the final tree response, the authors introduced a differential stochastic decision which assigns for each leaf l a probability of a sample reaching that leaf π_l, and μ_l defines the probability of the performed path by the sample until reaching that leaf. Fig.3.15 gives an example of μ_l calculation. The final tree decision $\mathbb{P}(y|x, \pi, \Theta)$ expressed in Eq.3.12 is rewritten as follows:

$$\mathbb{P}(y|x, \pi, \Theta) = \sum_{l \in \mathcal{L}} \pi_l \mu_l(x, \Theta) \tag{3.14}$$

and the final decision forest is performed as in the neural decision one. The trees are learned by minimizing a loss function $R(\Theta, \pi)$ defined as follows:

$$R(\Theta, \pi) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} -\log(\mathbb{P}(y|x, \Theta, \pi)) \tag{3.15}$$

where $\mathcal{X}$ represents the global initial dataset. The optimization is performed in the same way as in the neural decision forest, Θ and π are optimized alternatively.

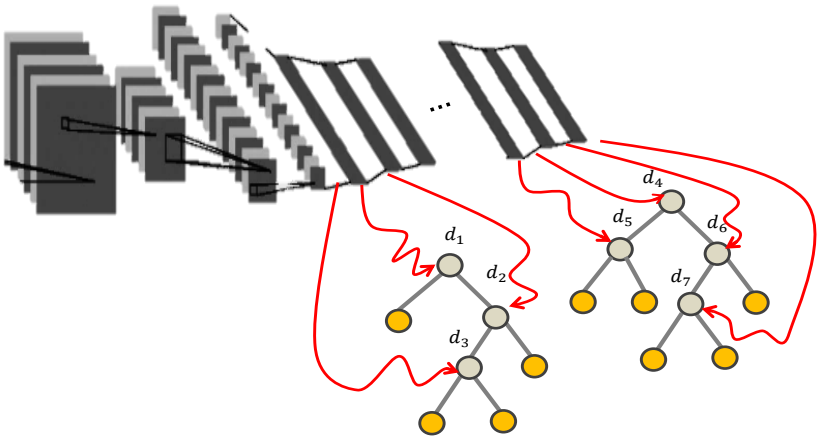


Figure 3.14: Instead of learning a novel representation at each node with MLP, deep forest performs a convolutional neural networks in end-to-end meaner (The fully connected layer is directly injected to the splitting nodes). The network and forest parameters are optimized alternatively.

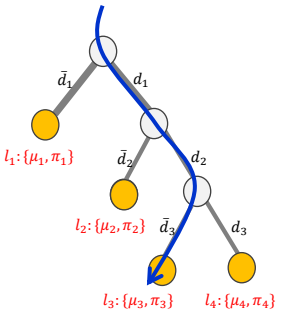


Figure 3.15: Differential and stochastic routing. Each node i performs a splitting operation via function d_i. The blue path shows an example of a sample x reaching the leaf l_3. The probability of the path l_3 encoded by μ_3 is equal to $d_1.d_2.\bar{d}_3$

3.5 Conclusion

In this chapter we presented the Random Forest framework by establishing mathematical notations. We detailed the training and testing modes of the ensemble of trees. We showed that this framework presents a high flexibility in handling different tasks depending on the nature of the input data. We performed different experiments to compare Random Forest to some alternative algorithms. For a classification problem, we noticed that Random Forest provides a prediction with uncertainty information, unlike SVM which produces a hard classification prediction. With continuous output, we compared the framework to the GP. Facing with multi-modal distributions, Random Forest presents better results. For a density estimation task, we performed a comparison between KNN and GMM. We noticed

that both models capture nicely the different distributions across the experiments. Nevertheless, Random Forest presented less artifacts. Even if the comparison experiments are based on relatively low dimension data, we provided real application with successful results of Random Forest in computer vision and medical image analysis. In the last section, we highlighted some recent ameliorations to overcome the limitation of the initial algorithm. Some modifications aim at randomness at node level which produces extremely randomized trees and ferns. Some solutions changed the prediction model part of the initial algorithm, for instance by combining both regression and classification information with Hough trees or by making the ability of introducing prior knowledge with conditional trees. A last modification is performed to enhance the discriminative ability of the trees by learning novel data presentation at each node with neural and deep trees.

According to the following points:

- Several promising results in computer vision and medical image analysis, highlighted in the previous chapter, are based on Random Forest algorithm;
- The previous discussed experiments, which compare Random Forest to alternative techniques, showed the great potential of the tree in terms of generalization and robustness;
- The nature of our task in terms of non linearities and formulation, meets some similar problems which are already solved using this technique;

we decided to treat all the problems, that we formulate as learning ones in this thesis, with this algorithm.

Chapter 4

Feature-based versus semi appearance-based approach

Contents

In this chapter we compare the potential of two recent approaches of automatic gaze estimation, feature-based and semi appearance-based approaches. The efficiency of the estimation of the first system is directly and strongly linked to the feature points localization accuracy. The efficiency of the second one depends on the robustness of gaze learning in frontal configuration and on the accuracy of the head estimation which allows the geometric correction yielding to the final estimation.

To achieve an accurate evaluation of the two approaches, we build two systems based on each strategy. We will discuss in the following sections the components of each system pointing the limitations which allow us to establish an objective comparison.

4.1 Feature-based approach

As mentioned previously, the main challenge in this approach is estimating accurately user eye key points locations and head pose. To build our feature-based system

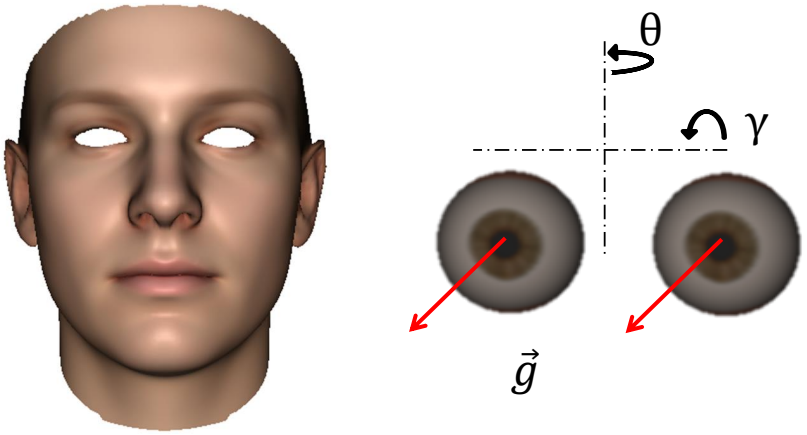


Figure 4.1: Visual illustration of the gaze vector $\vec{g}$. θ and γ angles represent the horizontal and vertical rotations respectively.

we use a 3D eye model to determine the 3D optical axis and infer the 3D visual axis. For this, we robustly estimate user head pose parameters and eye pupil locations using an ensemble of randomized trees trained with an important annotated training sets. Firstly, we project the eye pupil locations in the world coordinate system using the sensor intrinsic parameters. Based on a one-time simple calibration by gazing a known 3D target under different directions, the 3D eyeball centers are determined for a specific user for both eyes yielding to the determination of the visual axis expressed with two angles (θ, γ) (Fig. 4.2 shows the meaning of these angle according to the gaze vector g).

4.1.1 System architecture

Fig.4.1 shows an overview of our system composed of four components, each component can be described as follows:

- **Input** We grab the RGB and depth map at (1280×960) and (320×240) resolutions respectively at 15 fps (the fps is a hardware limitation of the sensor at these resolutions). Using the known Kinect intrinsic parameters and a predefined rigid transformation between the RGB and depth sensors, each depth value can be projected in 3D using the pinhole model as follows:

$$\begin{cases} x_d = \dfrac{d.(u_d - c_x)}{f_x} \\ y_d = \dfrac{d.(v_d - c_y)}{f_y} \\ z_d = d \end{cases} \tag{4.1}$$

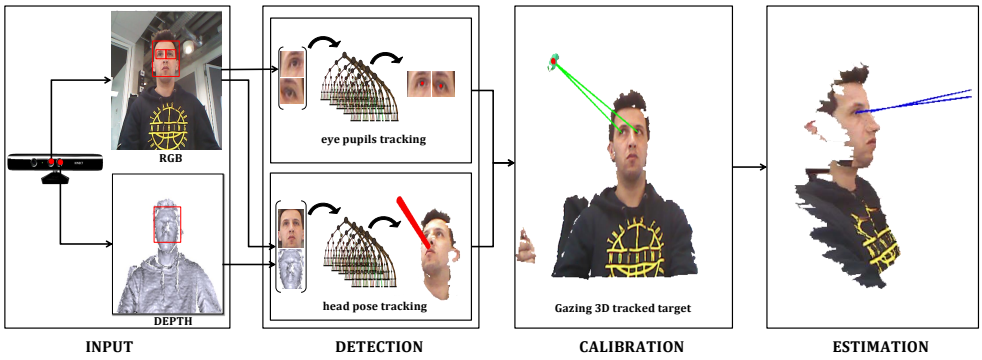


Figure 4.2: Overview of our approach. Four principal blocks can be distinguished. *Input* block describes the data grabbed from the depth sensor. *Tracking* block illustrates two global components, head pose and eye pupils estimation respectively using RGB-D cues. Using the computed information from the previous blocks, *Calibration* fixes for a specific user some parameters related to the eye geometry (performed by gazing an known target in 3D). Finally, *estimation* block gives gaze vectors for each eye.

where d represents a depth value with its coordinates $\{u_d, v_d\}$ and sensor intrinsic parameters $\{f_x, f_y, c_x, c_y\}$. (x_d, y_d, z_d) represent the final 3D projections. To produce a textured mesh as illustrated in Fig.4.1, a rigid mapping between RGB and depth sensor has to be established.

- **Detection** We use an ensemble of trees to train our head pose $\mathcal{T}_{head}$ and eye pupils $\mathcal{T}_{pupils}$ able to accurately and robustly estimate these parameters $\{(R,T),(u_p,v_p)\}$, where R and T represent the head rotation and translation matrix respectively according to the sensor coordinates space, u_p and v_p correspond to the pupil coordinates in the image space. We will discuss and evaluate in details these two components in the next sections. The detection block in Fig.4.1 describes the final estimations (red circles as eye pupils locations and red cylinder as head pose orientation).

- **Calibration** To compute the eyeball centers C represented in Fig.4.3, we assume a known target gaze point $G = (x_G, y_G, z_G)$ as illustrated in the calibration part of Fig.4.1. When the user is focusing at G, the angle between the optical axis $\overrightarrow{C_pP}$ and the visual axis $\overrightarrow{C_pG}$ would be α which is a constant value. [Guestrin 2006] describes an additional relation between the two axes as follows:

$$\frac{\overrightarrow{C_pG}.\overrightarrow{C_pP}}{||\overrightarrow{C_pG}||.||\overrightarrow{C_pP}||} = \cos(\alpha) \tag{4.2}$$

As the distances K_0 and K are constant, a relation between C and C_p can be

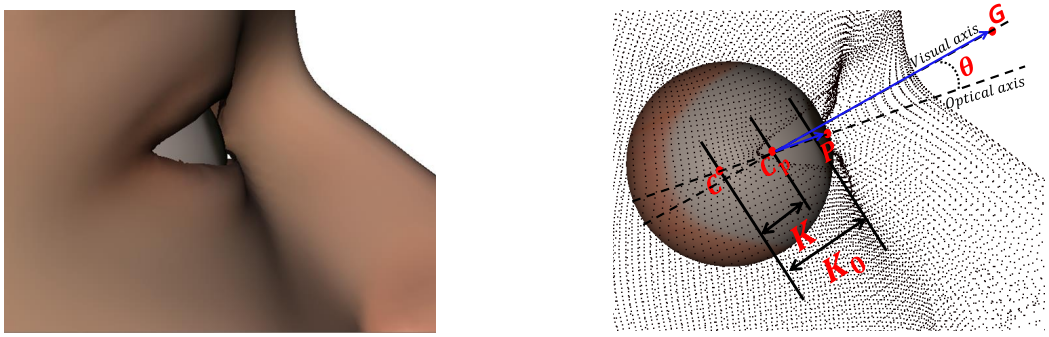


Figure 4.3: 3D eyeball model used in our method. C, C_p and P represent eye, cornea and pupil centers respectively. For human eye, $[CC_p]$ and $[CP]$ distances are constants. Visual and optical axes are represented in blue connecting $C_p - G$ and $C_p - P$ respectively. The dotted bow represents the angular relation between optical and visual axis (only vertical angle θ is illustrated here)

established as follows:

$$C_p = C + \frac{K_0}{K}(P - C) \tag{4.3}$$

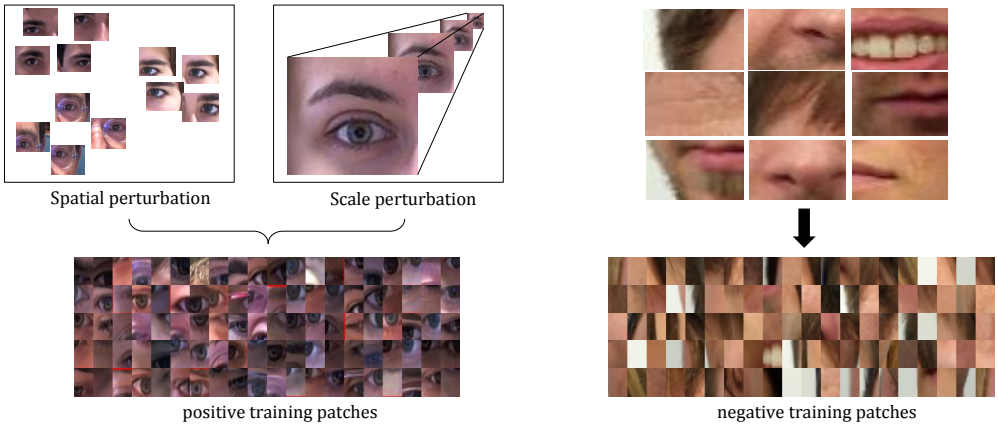
Using Levenberg-Marquardt optimization, the non-linear Eq.4.3 can be solved. By using the Eq.4.4 the eyeball center can be initialized at C_0 and transformed to the Kinect coordinate system as follows :

$$C = R.[C_0] + T. \tag{4.4}$$

- **Estimation** Knowing eyeball center C and the pupil P at each frame, cornea center C_p can be calculated. Thus, the optical axis can be estimated. By adding the constant angles values, the visual axis can be calculated and the gaze vectors can be expressed as vertical and horizontal angles (θ, γ) for each eye. As illustrated in Fig. 4.1, the estimation is represented in blue line for each eye.

4.1.1.1 Hough random forest for eye pupil localization

We detail here our method for eye-pupil localization in 2D images acquired from a simple uncalibrated monocular camera. We train an ensemble of trees able to learn the spatial relation between pupil image appearances and their corresponding 2D locations from a set of training samples. Our trees are trained in a way that they are able to capture both regression (estimating the 2D offset distance between a patch center and the hypothetic pupil location) and classification (predicting a patch class, positive or negative according to belonging to the eye image or not) from the training set. The final estimation is performed using a voting space (Hough image) which groups all the trees outputs. The 2D pupil location corresponds to the maxima



(a) (b)

Figure 4.4: Training data for eye pupil localization. (a) describes the final positive patches obtained after a spacial and scale perturbation on eye images. (b) shows some negatives training patches extracted from facial images with opposite to the eye regions.

in the voting space. To handle scale variations, we perform a pyramidal strategy by building an image pyramid at each testing step and the corresponding Hough image. We give more details about training data, learning and testing process in the following sections.

- **Training data:** Instead of learning of the whole images, our trees learn on a set of patches which represent a set of small groups of nearby pixels. We collect our training set $\{\mathcal{P}_i = (\mathcal{I}_i^o, y_i, c_i)\}$ from the databases [Villanueva 2013] and [Weidenbacher 2007] discussed previously where:

  - $\mathcal{I}_i^o$ represents the extracted visual features vector from a given patch $\mathcal{P}_i$ where o defines the feature channel, we used 1 channel namely the gray scale intensity of the eye regions. Theses regions are extracted using anthropomorphic relations after performing a face detection.
  - y_i describes the offset vector stretching from the patch center to the pupil center. This variable describes the regression information.
  - c_i represents the class of the patch, c_i equal to 1 if the patch is positive (extracted from pupil images), or c_i equal to 0 if the patch is negative (extracted from background images). For all the negatives patches, y_i is set to zero.

  To enhance the generalization of our trees, we introduce some perturbations in the extracted regions as [Markuš 2014] in scale with $[+30\%, -30\%]$ and in 2D pupil location by $[-25\%, +25\%]$ from the original. We collect $10k$ perturbed eye region samples from which we extract 50 patches of a fixed size (16×16)

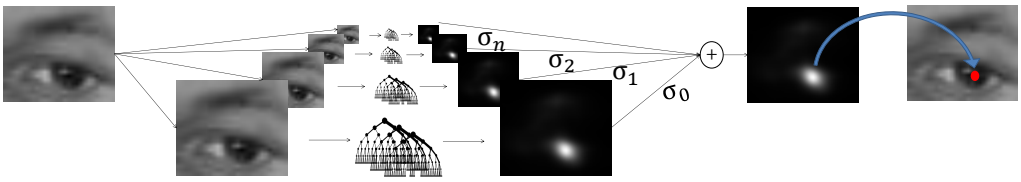


Figure 4.5: The pyramidal estimation strategy of our forest. Given a testing sample, we perform a pyramidal image using four scales, a set of patches is extracted from each scale and processed by the trees yielding a voting map per scale $\mathcal{H}_k$. A linear weighted combination of all these maps yields to a final Hough image $\mathcal{H}$. Finally, the 2D pupil location corresponds to the global maxima in this space.

per sample. Thus, we obtain $500k$ positive images. According to our problem, we extract a set of negative patches from regions belonging to the face but different from the eyes regions, we collect a total of $400k$ negative images. Our global training dataset consists of $900k$ while [Markuš 2014] learned on a corpus of 6M samples. Fig. 4.4 shows some examples from the training data.

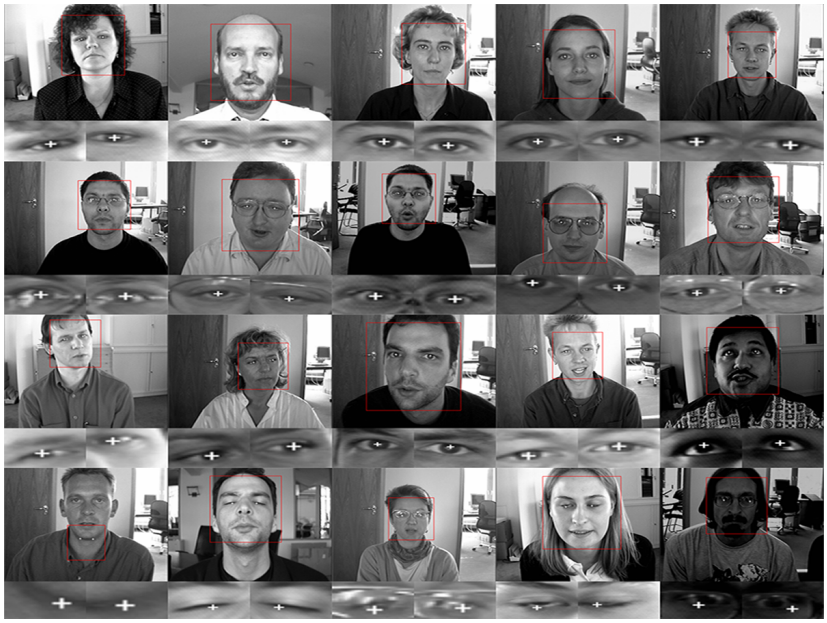
- **Training:** each tree χ in the forest $\mathcal{T}_{pupils} = \{\chi_t\}$ is trained on a set of training patches randomly selected from the global set $\{\mathcal{P}_i = (\mathcal{I}_i^o, y_i, c_i)\}$. At each splitting node, we define the binary test $h(\mathbf{x}, \phi)$ discussed in the previous chapter as follows:

$$\begin{cases} 1, & if \;\; \mathcal{I}_i^o(x_1, y_1) - \mathcal{I}_i^o(x_2, y_2) \leq \tau \\ 0, & otherwise \end{cases}$$

with $\psi = (x_1, y_1, x_2, y_2, o)$. The expression $(\mathcal{I}_i^o(x_1, y_1) - \mathcal{I}_i^o(x_2, y_2))$ represents the difference of intensity between two locations (x_1, y_1) and (x_2, y_2) in the channel o (in this case o represents the grayscale intensities). To supervise each tree and find the optimal ϕ^*, we maximize the information gain defined in Eq.3.5 and minimize the distance defined in Eq.3.6 at each node alternatively (randomly selected) until reaching the leaves. We fix the trees depth to 15 and generate for each splitting node a total of $10k$ binary tests. Each leaf l stores the following information:

  - $p(c|l)$ captures the probability of each class in the reached leaf l.
  - $\mathcal{N}(y_l, \bar{y}_l, \Sigma_l^y)$ represents the Gaussian distribution of all the offset vectors reaching the leaf l. $\bar{y}_l$ and Σ_l^y represent the mean and the covariance of the offset vectors respectively .
  - $\{y_i\}_{i \in l}$ represents the set of all the offset vectors reaching the leaf l.

- **Testing:** given an unseen sample, first we build an image pyramid. For each scale of the image pyramid, we extract a set of fixed size patches. Each patch is passed through all the trees of the forest. Each tree in the forest processes



(a)

Figure 4.6: Results of our method on BioID database. First row: good estimations on images under favorable conditions. Second and third rows: robust estimations under some unfavorable conditions. Last row shows some typical bad estimations.

the patch using all fixed binary tests. Using all the reached leaves, we build the Hough image $\mathcal{H}_s$ for each scale from the pyramid as a voting space. We project the set of pupil location candidates by adding all the offset vectors $\{y_i\}_{i\in l}$ to the each patch center y'. For a single tree, the candidates are represented as the sum of a Dirac weighted by the probability of belonging to the eye $p(c=1|l)$ in the reached leaf. Then, we average all the outputs over the forest. For a given number Ω of patches extracted from a given image from the pyramid, the Hough image $\mathcal{H}_k$ is represented as follows :

$$\mathcal{H}_k = \sum_{\mathcal{P}\in\Omega}\left(\sum_{t\in\mathcal{T}}\left(\sum_{i\in l}\frac{p(c=1|l)}{|\mathcal{T}|.|l|}\delta(y_i+y')\right)\right) \tag{4.5}$$

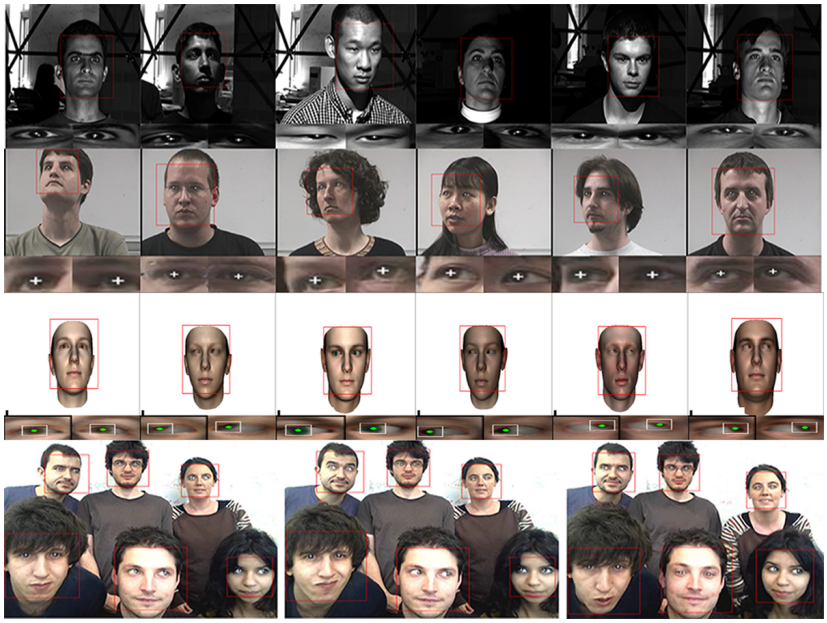
All the non-informative leaves presenting a high variance defined as $trace(\Sigma_l^y)$ and a low probability $p(c=1|l)$ are discarded.

Finally, we aggregate all the Hough images (resized to the same size) in a global Hough image $\mathcal{H}$

$$\mathcal{H} = \sum_{k\in Sc}\sigma_s\mathcal{H}_k \tag{4.6}$$

where:

$$\sigma_k = \frac{max(\mathcal{H}_k)}{\sum\limits_{k\in Sc} max(\mathcal{H}_k)} \tag{4.7}$$



(a)

Figure 4.7: Qualitative successful results on different databases. First row: Yale-extended database. Second row: Pointing 04. Third row: synthetic images and last row is multi-users images.

S_c defines the number of scale in the pyramid, we use 4 scales: 1.5, 1.0, 0.5, 0.25 according to the original image respectively. Fig. 4.5 illustrates an overview of the testing step and shows the mapping between the global maximum in the voting space and the pupil location in the 2D test sample space.

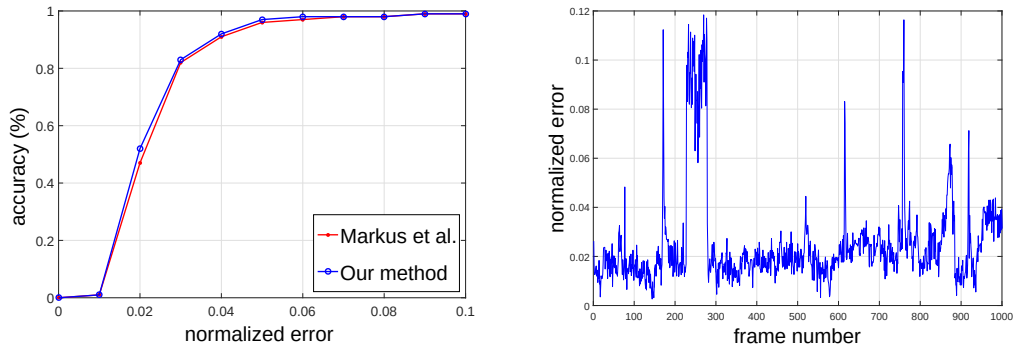
- **Results:** we provide here quantitative results of our experiments on still images and videos. Then we discuss the effect of some forest parameters on the estimation.

  To evaluate our method on still images, we compare it to the state-of-the-art on the BioID database. It contains 1521 annotated gray-scale images. BioID is considered among the most challenging databases for pupil localization due to its significant variations in terms of head pose variations, scale and illumination conditions. Like the majority of pupil localization algorithms, the metric introduced by [Jesorsky 2001] is used. It is defined by :

$$e = \frac{max\{D_L, D_R\}}{D} \tag{4.8}$$

  where D_L and D_R represent the Euclidean distances from the estimated pupil locations to those in the ground truth for left and right eyes. D is the Euclidean ground truth distance between right and left pupil locations.

  Tab.4.1 shows the comparison of our method with the state-of- the-art according to Eq.4.8. It represents the percentage of correct estimations for the given



(a) (b)

Figure 4.8: (*a*) Comparison of the accuracy curve between our method and [Markuš 2014] for the talking video database. (*b*) distribution of the normalized error over 1000 frames.

threshold(we use the same values provided by [Timm 2011], [Valenti 2008] and [Markuš 2014]).

Methods	$e \leq 0.05$	$e \leq 0.10$	$e \leq 0.15$	$e \leq 0.25$
[Jesorsky 2001]	38.0	78.8	84.7	91.8
[Timm 2011]	82.5	93.4	95.2	98.0
[Valenti 2008]	84.1	91.0	94.0	96.6
[Markuš 2014]	89.9	97.1	–	99.7
Our method	**91.3**	**97.9**	**98.5**	**99.6**

Table 4.1: Comparison of pupil 2D localization on the BioID database. The authors of [Markuš 2014] do not provide the result for $e \leq 0.15$ but we point it out as an empty case.

- $e \leq 0.25$: Usually used for face matching, it corresponds to the distances between the pupil center and the eye corner. It indicates that the estimation belongs to the eye region which represents a low level of precision. The majority of methods gives approximately the same results.
- $e \leq 0.15$ and $e \leq 0.10$: Our method yields better results compared to [Valenti 2008] and [Timm 2011]. The circularity of the pupil which represents a strong assumption of the last two methods is not guaranteed due to significant changes in the head pose. The presence of eye images under head pose variations in our training data makes our method robust to this kind of constraint.
- $e \leq 0.05$: Corresponds to a high level of precision in estimation. It indicates very low distances from the pupil center. Compared to [Markuš 2014]

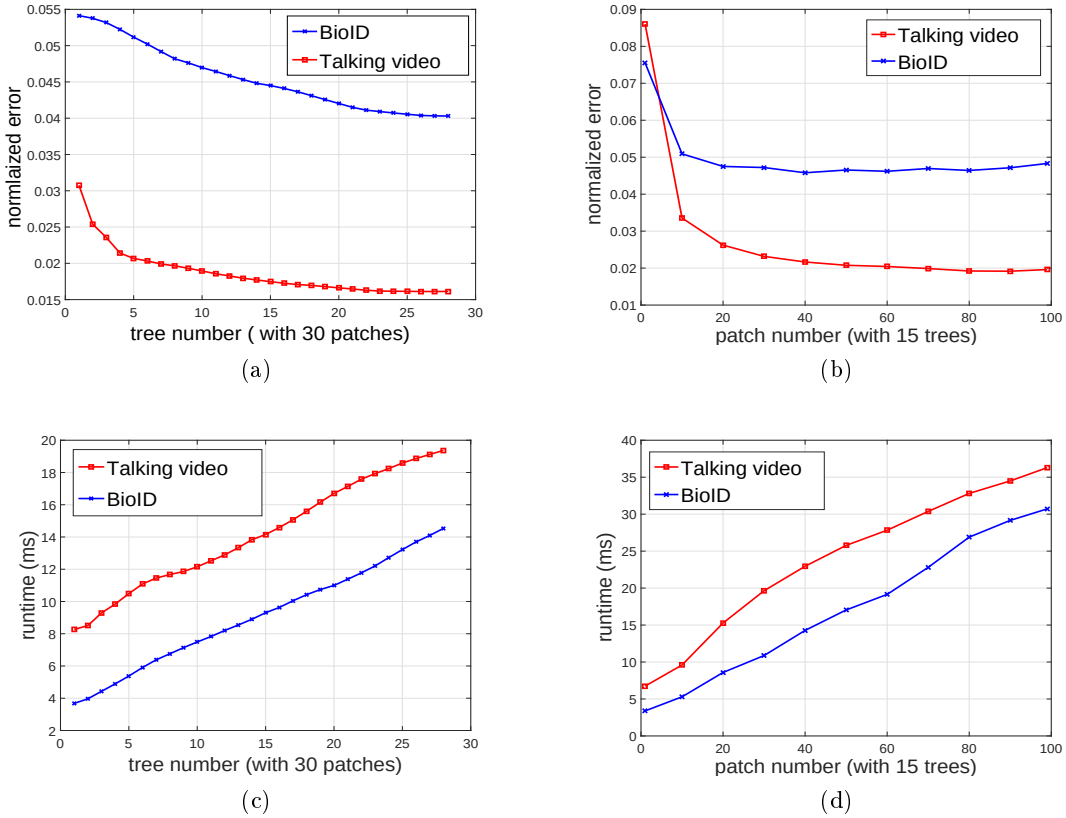


Figure 4.9: The forest parameters effect on the pupil location estimation on 500 frames from the BioID database labeled in blue and the talking video database labeled in red. a) The normalized error behavior under trees number variation with the number of patches extracted fixed to 30. b) The normalized error variation as a function of the number of patches extracted when the number of trees is fixed to 15. c) and d) represent the time needed to regress the output of all 500 frames under number of trees and the number of patches variations extracted respectively.

our method gives better results. The projection on Hough space implies an extension in the regression space of the forest. In addition, the absence of some typical examples like the presence of glasses in the training data in [Markuš 2014] paralyzes this method in some scenarios.

Fig. 4.6 and Fig. 4.7 shows a visual a illustration of 2D pupil estimation. The failures represented in the last row of Fig. 4.6 can be explained by the following:

- The failure of the face detector as shown in the first example of the last row which distorts the research area.
- The eyes appearance distorted by highlights on the glasses, dark illumi-

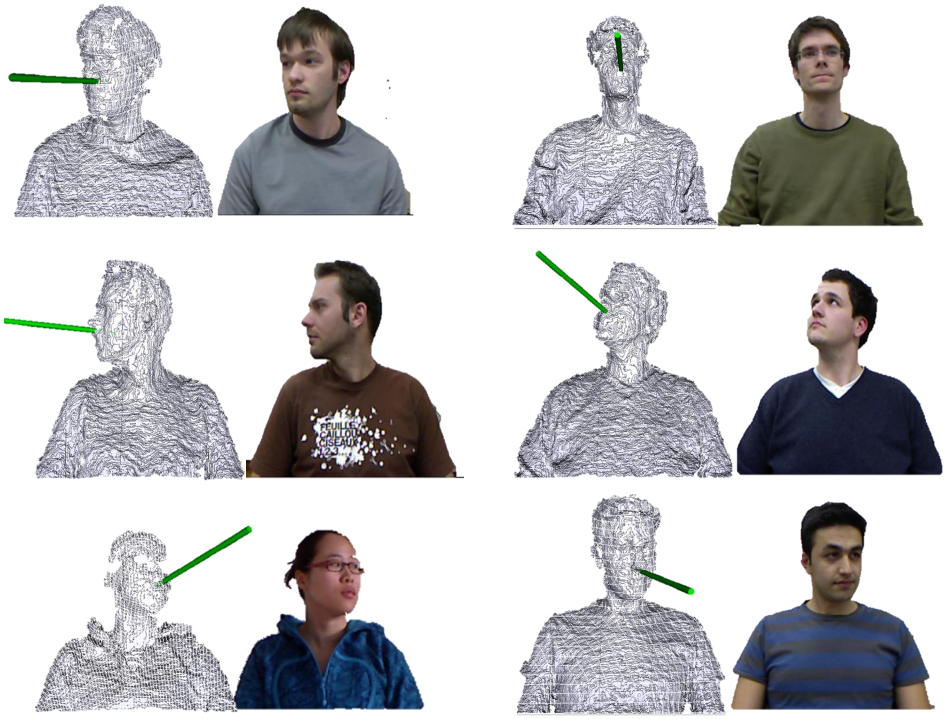


Figure 4.10: Examples of labeled head pose samples extracted from the Biwi database. For each user an RGB-D cue is provided with the corresponding annotation (illustrated by a green cylinder)

nation or eye closure.

For video scenarios, as done in [Markuš 2014], we evaluate our method with the public database Talking Face video. It contains 5000 frames of a person engaged in a conversation. A specific active appearance model [Cootes 2001] is trained to annotate the frames accurately. The forest trained in the previous section and the metric of Eq.4.8 are used for the evaluation of our method.

We average the normalized error over all the frames. We obtain a mean error equal to 0.190. Because the authors of [Markuš 2014] did not provide numerical results for their method, we tried to reproduce their accuracy curve at best and compare it to our approach. Fig. 4.8a illustrates the comparison indicating that both methods give an estimation belonging to the pupil-radius ($e < 0.10$) over all the frames. The closure of eyes and the wrong annotations in some frames as shown in Fig. 4.8b explain the peaks on the distribution of the normalized error.

Our method is controlled by some parameters. The previous experiments were performed under optimal values of these parameters.

– The number of trees used for the estimation. Fig. 4.9a illustrates the variation of the normalized error defined in Eq.4.8 for 500 images from

the BioID and talking video databases under different values of forest size. The error decreases by increasing the number of trees used for both databases (note the apparent gap between the two curves due to the different resolution of the images in the two databases). The normalized error is reduced by approximatively 30% compared to the initial value (from 0.055 to 0.040 for BioID and from 0.032 to 0.170 for the talking video) which is the result of output smoothing by the different trees. We noticed that, using more than 25 trees does not produce more precision, so we fix the optimal forest size to 25. Fig.4.9c shows the time in seconds needed to process the 500 frames under different sizes in the forest approach. The use of 25 trees gives an average fps of 30.

- The number of patches extracted from the testing image. Fig.4.9b represents the variation of the normalized error under different numbers of patches used for the estimation. The normalized error is reduced approximatively by 75% for the talking video database (from 0.082 to 0.02) and 45% for the BioID database (from 0.078 to 0.044). By increasing the number of patches, the trees get more information about the input test image which consequently gives more accurate estimations. In our experiments, according to the dimension of the image test (80×70), we noticed that 35 patches cover approximatively all the input information. Fig.4.9d shows the time needed to process 500 frames for different numbers of extracted patches.
- The maximum variance which is represented by the trace of the covariance of the offsets reaching each leaf is fixed to 800 and the probability $p(c/w)$ is fixed to 0.7. These values seem to provide good estimation results, a variance of 800 defines a voting area of approximatively (20×20) from the pupil center. The patch size of (16×16) gives an acceptable appearance which allows a good discrimination and generalization of the forest during the estimation.

4.1.1.2 Regression Forest for head pose estimation

According to the promising results obtained in [Fanelli 2011] with Random Forest algorithm, we decided to train our forest to robustly estimate the head pose parameters. As done for the eye-pupil localization component, we develop the training and testing steps.

- **Training data:** As done for pupil localization, we extract a set of patches $\{\mathcal{P}_i = (\mathcal{I}_i^o, y_i)\}$ from Biwi database developed in [Fanelli 2013] described in Fig.4.10 (Each head pose sample is a couple of RGB and depth image, the green line represents the ground truth) where:
  - $\mathcal{I}_i^o$ defines visual features vector extracted from $\mathcal{P}_i$, with $o = 2$ ($\mathcal{I}_i^0$: represents the depth values and $\mathcal{I}_i^1$: describes the gray-scale intensities).

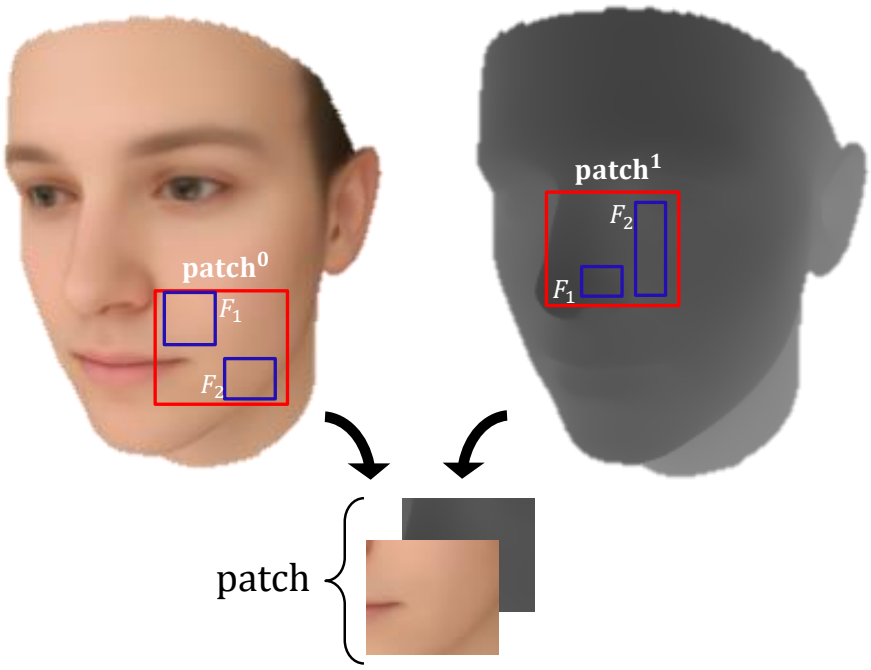


Figure 4.11: Patch extraction from RGB-D sample (red rectangles). The blues rectangles describe a possible choice of the binary test difference.

  - y_i describes the head-sensor rigid transformation with 6 parameters $\{T_x, T_y, T_z\}$ as the translation parameters and $\{roll, pitch, yaw\}$ as the rotation parameters.

- **Training:** To split the data at each node, we define at each node the following binary tests $h(\mathbf{x}, \phi)$ as follows:

$$\begin{cases} 1, & if \ \frac{1}{|F_1|} \sum_{q \in F_1} \mathcal{I}_i^o(q) - \frac{1}{|F_2|} \sum_{q \in F_2} \mathcal{I}_i^o(q) \leq \tau \\ 0, & otherwise \end{cases}$$

with $\psi = (F_1, F_2, o)$. Unlike in pupil localization, a normalized difference is performed on a sum of the pixels within two rectangles F_1, F_2. Fig.4.11 illustrates a path example with two channels. To supervise each tree, we maximize the regression information gain defined in Eq.**??**. By assuming the covariance matrix Σ a block-diagonal, the information gain can be re-written as follows:

$$E_j = \log(|\Sigma_a| + |\Sigma_b|) - \sum_{i \in \{\mathcal{L}, \mathcal{R}\}} \frac{|\mathcal{P}_j^i|}{|\mathcal{P}_j|} \log(|\Sigma_a| + \Sigma_b|) \tag{4.9}$$

where Σ_a and Σ_b encodes the covariances of the translation and the rotation parameters of the head pose respectively.

To train a total of 25 trees, $8k$ RGB-D samples are used, which represents $\frac{2}{3}$ of the global database. From each sample, 20 patches are extracted, providing an amount of $160k$ head pose RGB-D patches of a fixed size of (80×80). Each predictor is trained on 50% of the initial training set with a maximum depth of 15 and $20k$ binary tests.

- **Testing:** given an unseen RGB-D sample, we extract patches after performing a face detection. Each patch is passed through all the trees of the forest. Instead of performing the same strategy as done in pupil localization forest, we perform a non-parametric clustering of the votes. We used 5 mean-shift iterations with a spherical kernel to find the global maxima for both translation and rotation outputs. The leaves with high variance are non informative, so we discarded them.

  To evaluate the accuracy of our forest, we perform experiments following the same protocol as done in [Fanelli 2011]. We report the results of the head pose estimation errors on the remaining $\frac{1}{3}$ of the Biwi database. Tab.4.2 illustrates the errors related to two angles of the head pose under two user-sensor distances. Under 75 cm from the sensor, the two methods achieve approximatively similar accuracy, while our method slightly outperforms the method in [Fanelli 2011] for *yaw* and *picth* respectively. Indeed, for large distances, depth cue suffers from real noise presence, using grayscale channel conjointly with depth compensates this trouble, yielding better results.

Methods	Mean Absolute Error			
	75 cm		150 cm	
	Yaw	Pitch	Yaw	Pitch
[Fanelli 2011]	5.7°	5.1°	8.7°	9.7°
Our method	6.0°	5.2°	8.3°	9.2°

Table 4.2: Comparison of our head pose estimation to [Fanelli 2011] across two user-sensor distances, 75 cm and 150 cm respectively. .

Fig.4.12 shows some successful head pose estimations using our forest $\mathcal{T}_{head}$ on unseen RGB-D samples acquired from the Kinect.

4.1.2 Gaze estimation results

In our experiments some parameters related to eyeball geometry are fixed beforehand. The constants K_0 and K inside the eyeball are fixed at the average human values to 5.3cm and 13.1cm respectively. The horizontal and vertical angles between visual axis and optical axes are fixed to 5° and 1.5° respectively as done in [Guestrin 2006]. We calibrate the eyeball for a specific user by solving the non-linear Eq.4.3 with 5 gaze samples recorded under different directions.

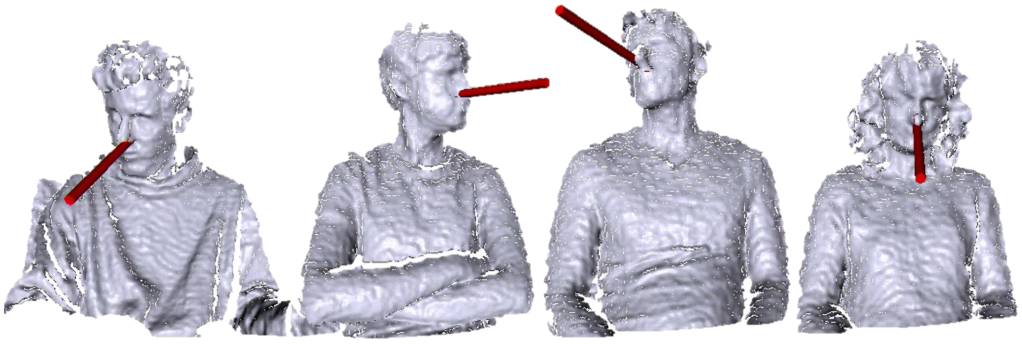


Figure 4.12: Head pose estimation based on random regression forest (established by the red lines). The depth images are acquired from the Kinect sensor using (320×240) resolution.

To evaluate our method, we design a target point represented by a green marker cap which can be easily tracked in 3D (based on color segmentation as done for the calibration step) moving in front of the user. We tested gaze estimation accuracy when the target is moving upward and rightward with two user-sensor distances (75 cm and 150 cm). Fig.4.13 shows the comparison diagram between ground truth and our estimation for the upward scenario, where only θ is changing while γ is changing for the rightward one for both eyes. As we can see our estimation is close to the ground truth. Comparing to [Mora 2012], our method gives better results and the average error remains below 5.5°. For 150 cm distance, RGB and depth image resolutions decrease significantly giving a less accurate head and pupils tracking, producing higher gaze estimation errors. Fig.4.14 shows the gap between estimation and ground truth. However errors are still acceptable (less than 7.5°). Despite robustness of our tracking component, the difference in RGB and depth resolutions (which is a hardware limitation) makes projection of the 2D pupil locations in the sensor coordinate system very sensitive and gives sometimes instable gaze vectors.

Methods	75 cm				150 cm			
	upward		rightward		upward		rightward	
	θ	γ	θ	γ	θ	γ	θ	γ
[Jianfeng 2014]	5.81°	2.68°	5.44°	5.19°	-	-	-	-
Our method	3.46°	4.09°	4.70°	3.15°	4.61°	5.53°	5.65°	4.16°

Table 4.3: Mean gaze estimation error across two user-sensor distances.

In Tab.4.3 we compare our method to [Jianfeng 2014] which used the same strategy. We report the mean gaze estimation error over the two eyes for the two directions θ and γ respectively.

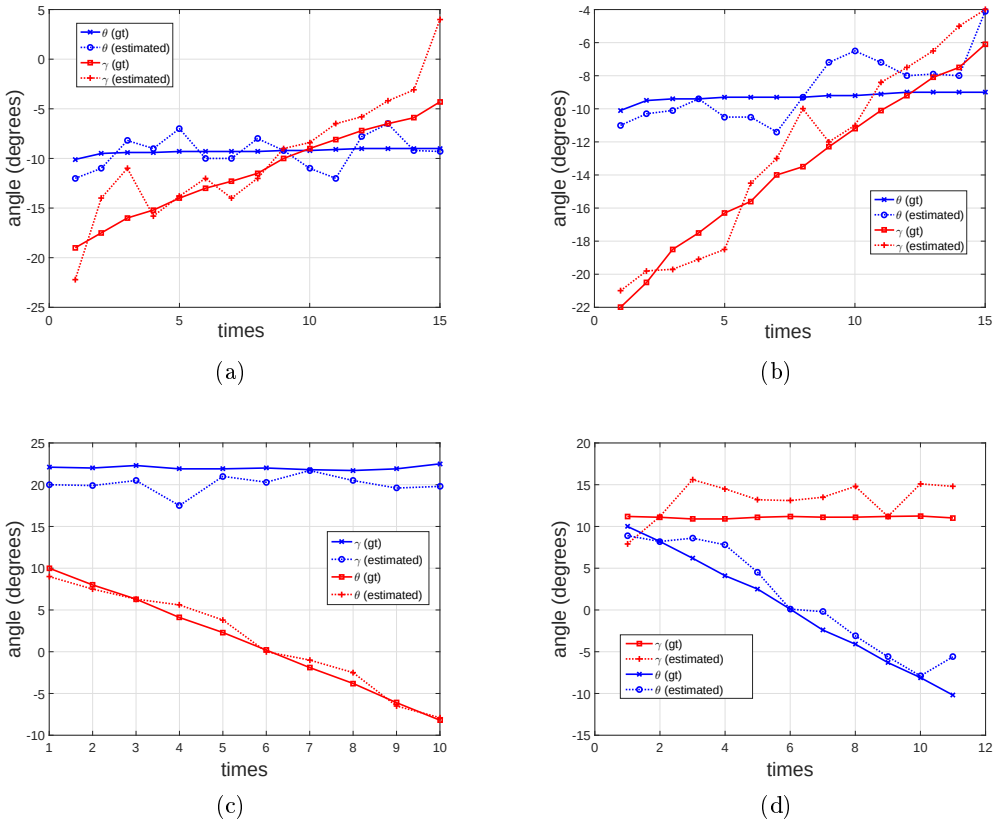


Figure 4.13: Gaze estimation error at 75 cm. (a) and (c) describe gaze estimation errors for upward and rightward moving of the target for right eye. (b) and (d) illustrate errors for left eye under the same scenarios.

4.2 Semi appearance-based approach

To overcome the feature points localization problem, this approach learns a direct mapping from the eye appearance to the gaze vector space. To achieve sufficient accuracy, the methods which are based on this approach decorrelate the head pose component from the global automatic gaze system. By assuming that the eye appearances are extracted from a frontal face manifold, a geometric correction based on the head pose parameters is performed to infer the final gaze information. The operation of building a manifold of frontal appearances is called frontalization. [Mora 2012] and [Lu 2011a] achieved promising results using this approach.
‘

4.2.1 System architecture

We present in Fig.4.16 our semi-appearance automatic gaze estimation represented with 3 global blocs. We exploit the multi-modality of the Kinect sensor by using

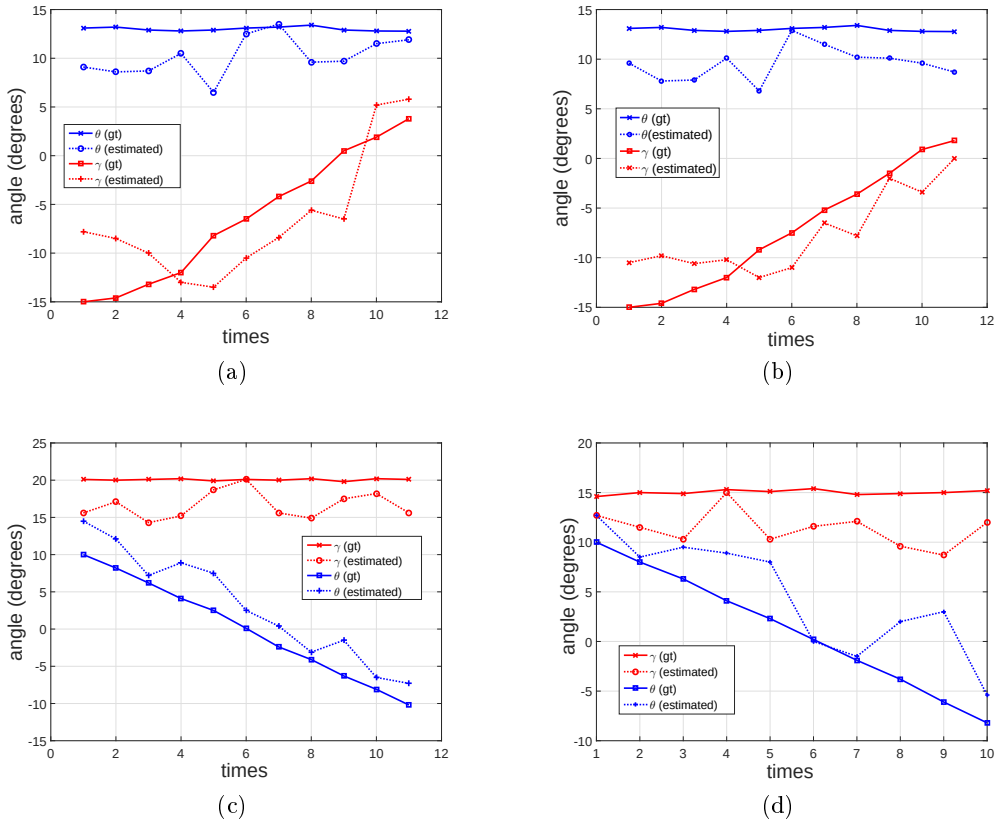


Figure 4.14: Gaze estimation error at 150 cm. (a) and (c) describe gaze estimation errors for upward and rightward moving of the target for right eye. (b) and (d) illustrate errors for left eye under the same scenarios.

RGB and Depth streams (at SXGA and VGA resolutions respectively). We build a 3D textured mesh as explained previously, which represents the principal input of our system. We perform a head pose normalization as done in [Mora 2012], instead of using a 3D face model to extract the eye regions. We perform a face detection and use anthropomorphic relations which represent a projection instance on a mapping function learned from frontal gaze samples. Unlike in [Lu 2011b], [Lu 2011a] and [Mora 2012] which use ALR (Adaptive Linear Regression) to capture the mapping function f between the eye appearance space and the gaze estimation, we train a set of tree predictors learned on a set of annotated training gaze samples. We will detail the frontalization and gaze learning step in the following sections.

4.2.1.1 Frontalization

Face frontalization describes the process of recovering the frontal views of faces by normalizing the pose in 2D or 3D. These recent years, several researches focus on this task according to its importance on many face analysis application such as gaze

Figure 4.15: Some results of the final gaze estimation using our feature-based system. The green lines define the ground truth and blue lines represent the estimation, the red point shows the tracked object.

estimation. [Ding 2015] gives a comprehensive survey.

Among the methods relying on the 2D images, we report the *piece-wise warping*. This approach normalizes the pose in a piecewise manner by projecting the face shape to a specific pose. By performing a Delaunay triangulation, each piece is assigned to a single triangle. To transform each triangle from the original to the target, a linear warping is generally performed as done in [González-Jiménez 2007] and [Gao 2009]. Instead of geometric assumptions, [Asthana 2009] proposed to learn the correspondence between facial landmarks in frontal and non-frontal configurations with a Gaussian Process Regression. Recently, [Ding 2015] used a generic 3D face model with predefined landmarks which are aligned to those of the 2D image with a compensation strategy to preserve the identity. Another way to perform such operation relies on *patch-wise warping* strategy. Introduced by [Ashraf 2008], the authors proposed to learn the optimal affine between a collection of patches from the frontal to the non-frontal domain using Lucas-Kanade algorithm. [Ho 2013] injected consistency at the overlapped pixels between nearby patches and proposed to globally learn optimal set of local warps using Markov random fields. A last way to frontalize in 2D is the *pixel-wise displacement*. Introduced by [Li 2012], the main idea of this approach consists in establishing a dense pixel-wise correspondence between images of different poses using optical flow as template displacement fields. Given a testing image, the frontal face is synthesized using a linear combination of these template fields.

In contrast to the approaches discussed above, some methods employ the 3D information to normalize the pose. The general principle consists in aligning the 2D face image with a 3D face model using some invariant facial landmarks locations. Then the 2D texture is mapped on the 3D model and rotate to a frontal pose, a final projection of the model is performed to obtain the 2D result image. The most famous and relevant method is the proposed approach in [Vetter 1998] and [Blanz 2003] which consists in aligning a semantic 3D morphable model to the single 2D image using a coarse fitting strategy (we will discuss this model in details in the next

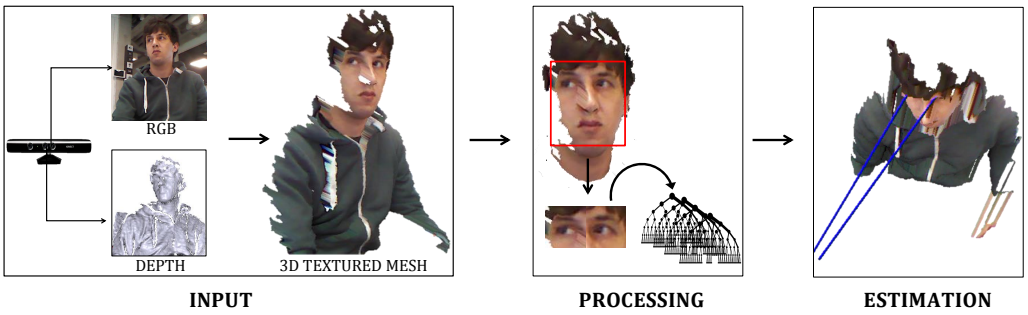


Figure 4.16: Our semi-appearance automatic gaze estimation system. As the previous system, we grab the multi-modal data from the Kinect sensor and build a 3D textured mesh. A processing step is performed to normalize the head pose to obtain a frontal view of the face to extract the eye regions RGB information. These regions represent the input vector of a learned forest which yields the final estimation.

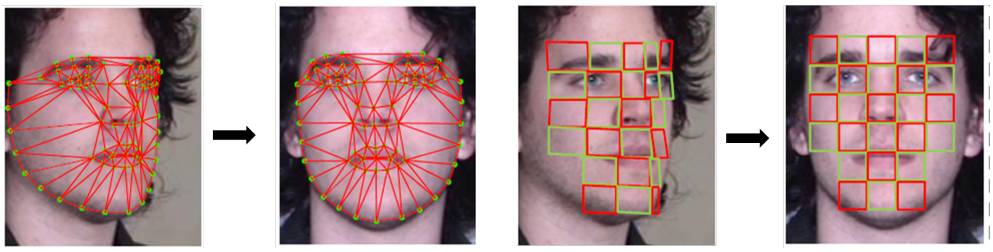


Figure 4.17: 2D frontalization.(a) describes the piece-wise warping method, each triangle is warped to get the final frontal 2D image. (b) instead triangulation, patches-wise warping method warps some extracted patches to reconstruct the frontal view.

chapter).

In our semi-appearance automatic gaze estimation, we perform a pose normalization using the inverse head pose parameters estimated by the forest $\mathcal{T}_{head}$. Assuming the following head-sensor rigid transformation:

$$A = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

the inverse transformation A^{-1} can be written as follows:

$$A^{-1} = \begin{bmatrix} R^{\top} & -R^{\top}.T \\ 0 & 1 \end{bmatrix}$$

Instead of using a 3D face model, we use the multi-modal data grabbed from the kinect (after establishing a rigorous calibration between depth and RGB sensors)

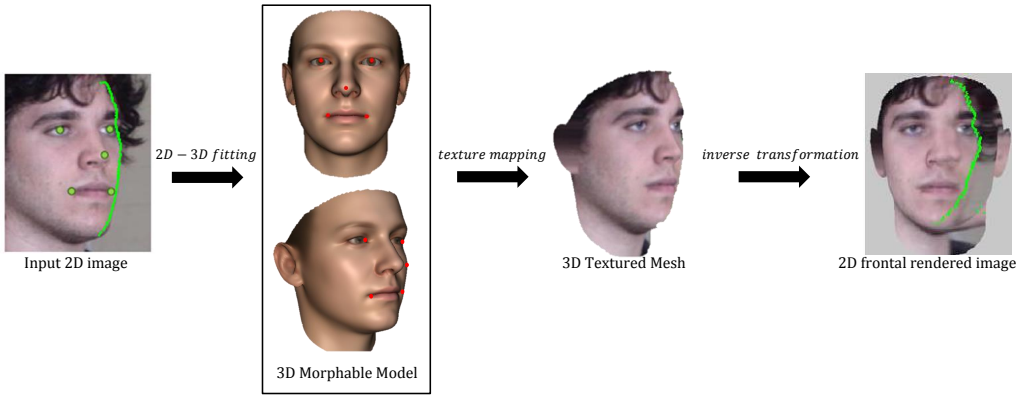


Figure 4.18: 3D frontalization. Given a 2D image and a 2D face model with the same landmarks respectively, a fitting 3D-2D is performed (align the projection of the 3D model to match with 2D image using the landmarks), the initial 2D texture is then mapped to the 3D model. The obtained textured mesh is then transformed using head pose parameters to a frontal view.

to build a textured mesh. Each vertex v_i in the global mesh $\mathcal{M}$ is transformed to v'_i using A^{-1}. The final frontal mesh $\mathcal{M}' = \{v'_i\}$ is projected in an orthographic manner to obtain the 2D frontal image.

To illustrate the importance of the pose normalization and its involvement in enhancing the performances in face analysis problems, we proposed to evaluate its influence on the face detection task. For this purpose, we first evaluate the face detection rate on the Biwi database and their correspondent frontalized images (using a ground truth head pose parameters). We perform the same procedure on our own frontalized samples based on estimated head pose parameters. A total of 7250 images from Biwi (representing 24 different participants) are used, and 2727 from our data (representing 42 participants). Fig.4.20 describes some face detection results with and without pose normalization. To quantify the involvement of the frontalization, we performed face detection using the well known method from [Viola 2004] under 5 different head pose configurations. Fig.4.21 reports the obtained results and details the head pose changes used during the experiment. These results strongly confirm the importance of this component in face analysis tasks such as gaze tracking.

4.2.1.2 Frontal gaze learning

Once the head pose normalization is performed, we detect the face and extracted eye regions from the obtained frontal 2D image. We collect a set of samples ($\mathcal{P} = \{\mathcal{I}_i, g_i\}$) where $\mathcal{I}$ encodes appearance of the eye image i as the grayscale values, g represents the gaze information as two angles (θ, γ).

To collect the gaze information, we established a simple setup which consists on

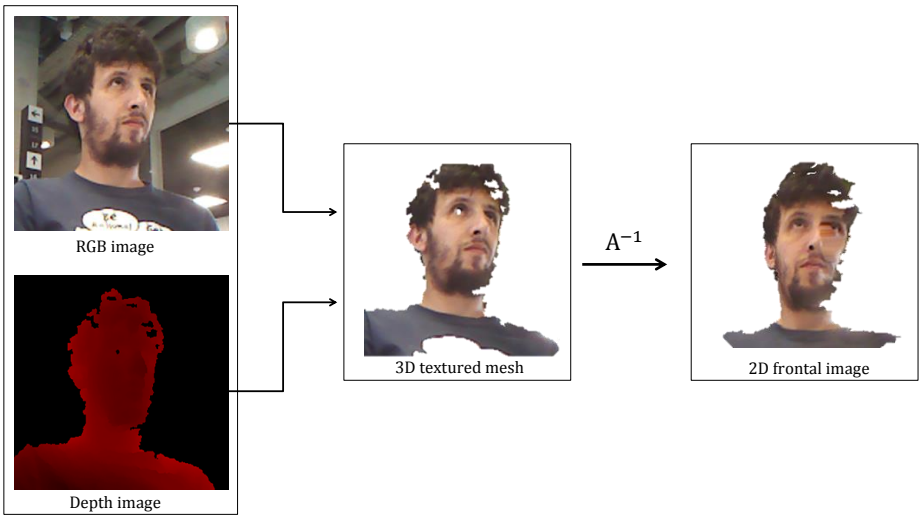


Figure 4.19: Our face frontalization. Using the multi-modal kinect data, we first build a 3D textured mesh. Then our forest takes the depth map as input and compute the head pose transformation A, each vertex from the 3D mesh is then transformed using the inverse transformation A^{-1} to obtain a frontal view.

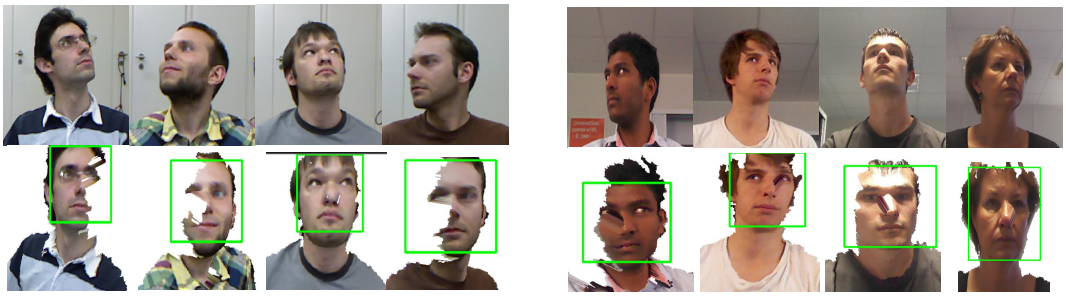
gazing at a moving point in a 2D screen. Knowing the sensor-screen rigid transformation and the Kinect intrinsic parameters, the 2D point can be projected and expressed in the Kinect world coordinates. We estimated the head pose of each participant and performed a frontalization. $g(\theta, \gamma)$ represents the existing angles between the 3D projection of the moving target (transformed with head translation and rotation in the frontal manifold) and a reference point extracted from a given eye region (in our case, we take the center of the extracted eye region). Fig.4.22 describes in details our protocol to collect gaze samples, and shows some eye images appearances (the apparent white artifacts are due to the frontalization).

4.2.2 Gaze estimation results

To evaluate our semi-appearance gaze system, we perform different testing gaze samples through 10 users using the same setup. We compare our method to a similar approach achieved in [Mora 2012] under 2 user-sensor distances, 75cm and 150cm respectively.

Table 4.4: Mean semi-appearance gaze estimation error across two user-sensor distances.

Methods	75cm		150cm	
	θ	γ	θ	γ
[Mora 2012]	6.6°	7.6°	-	-
Our Semi.app-based	6.9°	7.1°	7.2°	7.4°



(a) (b)

Figure 4.20: Face detection improvement using frontalization. (a) shows some examples from the Biwi database. We first search for a face on the original images with no results unlike frontalized images in the second row, each green rectangle indicates a face presence. (b) illustrates the same experience on our RGB-D samples.

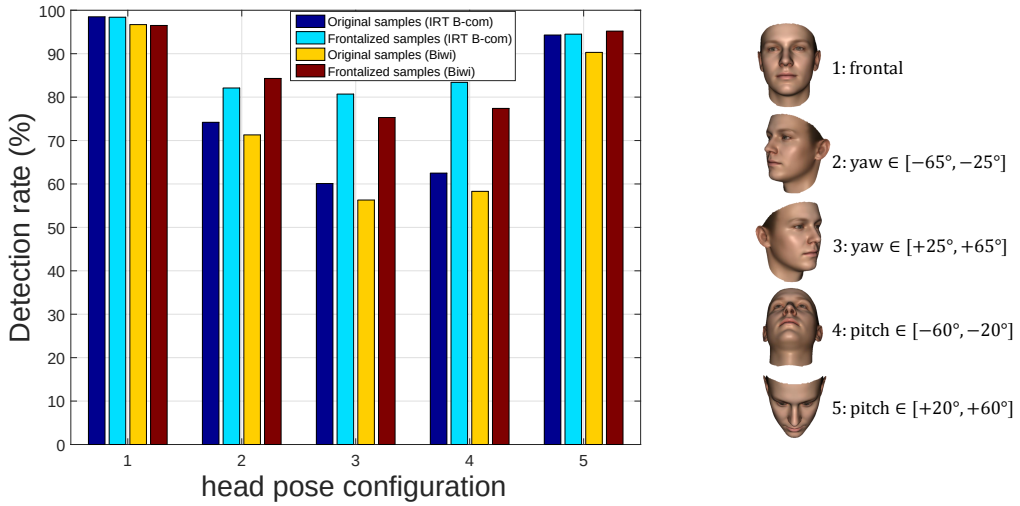
Tab.4.4 reports the gaze estimation errors. We take the best results from the method in [Mora 2012] through the participants, and the mean error over 10 users in our case. Our method gives slightly better results. As the frontalization step is applied using relatively the same head pose parameters in terms of accuracy, the difference can be explained by the robustness of our learned trees to capture the mapping between the corrected eye image appearance and the gaze information. We also notice that our system still gives slightly important errors for a larger user-sensor distance namely 150cm (the errors are not provided for this distance in [Mora 2012]).

4.3 Comparison

We have developed our feature-based and semi-appearance based approaches and have compared them separately to some state-of-art methods. To identify the relevance of each system, we achieve experiments using the same input data (same 10 participants) under identical environments conditions, we report in Fig.4.23a the gaze estimation errors performed by each system.

Fig.4.23a compares our two systems under frontal configuration with a user-sensor distance of 150cm. The feature-based system presents better results with mean errors of 5.6° and 5.7° on the two directions θ and γ respectively. The eye-pupil localization presents high accuracy under frontal configuration achieving good gaze estimation. The semi-appearance based system produces more important errors, 7.1° and 7.4° for the two directions due to the difficulty of generalization of our learned trees (especially for some participants such as 3,5 and 6 who wear glasses).

In Fig.4.24a we perform the same experiments with strong head pose changes. The semi-appearance based system presents better results with 8.2° and 8.1° as mean errors. Performing a frontalization producing a frontal view yields a consistent



(a) (b)

Figure 4.21: Face detection improvement under 5 different head pose configurations. (a) illustrates quantitative results of the face detection using Biwi and our RGB-D samples, we reported an improvement of 9.9% and 11.16% for our data and Biwi respectively. (b) details the range of each head pose configuration.

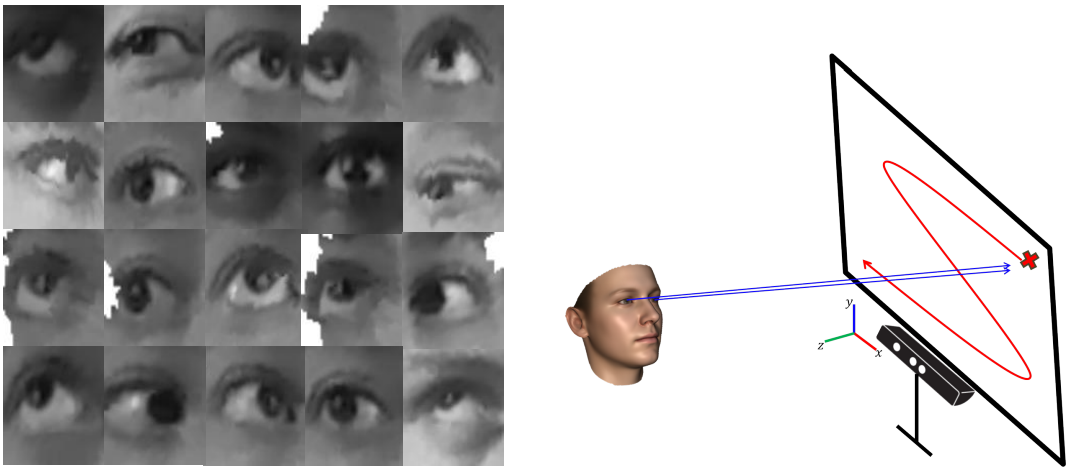
manifold for the obtained eye image appearances which gives acceptable results even with a high eye image appearance variability. Unlike the previous, the accuracy of feature-based system decreases drastically giving 9.6° and 9.8° for the two directions respectively. This behavior can be explained by the eye-pupil localization accuracy which decreases with important head pose changes giving a very instable projections in the Kinect coordinates system that produces very sensitive gaze estimations.

4.4 Conclusions

In this chapter, we built two automatic gaze estimation systems following two different approaches, feature-based and semi appearance-based approaches.

To build a robust feature-based system, we trained an ensemble of regression trees coupled with a Hough space voting to localize the eye-pupil center and projected it into the Kinect coordinate system. Using a one-time specific user calibration and the head pose parameters estimated with another trained regression forest, we obtained the visual axis for each eye. Our semi appearance-based system follows the standard pipeline with a frontalization step which consists in reconstructing at each frame the frontal face view. This approach aims at reducing the eye images appearance manifold across head pose changes and learns a robust mapping function. Our system use tree predictors to capture this mapping and produce visual axes.

We achieved different experiments to evaluate the accuracy of each system. The

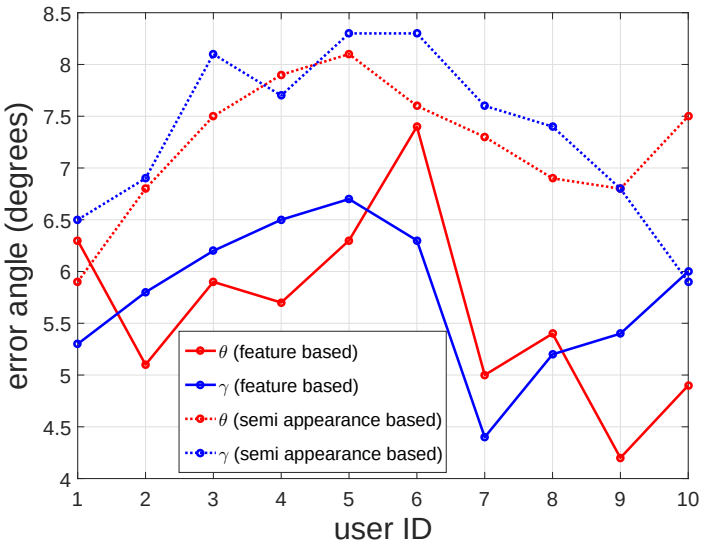


(a) (b)

Figure 4.22: Frontal gaze estimation learning.

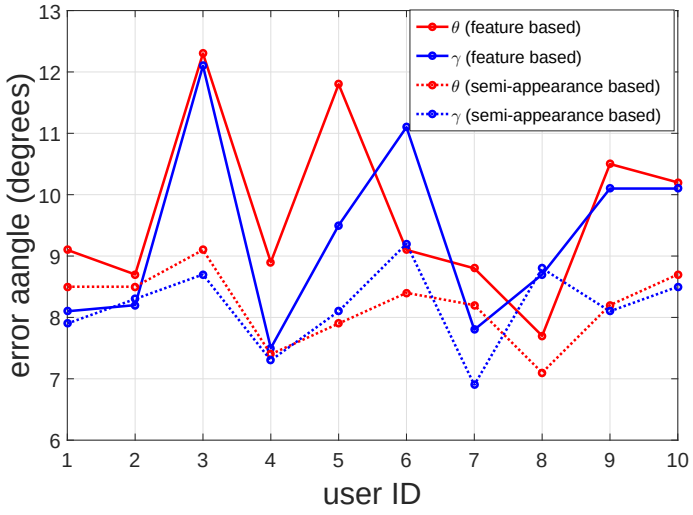
results show that our methods perform slightly better compared to the state-of-the-art but still present high gaze estimation errors in large user-sensor distances and strong head pose changes. In addition, we identify the advantages of each system across the different scenarios, we conclude that the semi appearance presents relevant behavior under head pose variations.

To overcome this limitation, one solution can be considered which consists in considering the head pose and eye appearance as a single block. We describe this approach as fully appearance automatic gaze estimation, we give more details in the next chapter.



(a)

Figure 4.23: feature-based and semi appearance system gaze estimation comparison under relatively favorable conditions with low head pose movements.



(a)

Figure 4.24: feature-based and semi appearance system gaze estimation comparison highly unconstrained environment with important head pose variations

Chapter 5

Fully appearance-based approach

Contents

The analysis and comparison of the feature-based and semi appearance-based approaches in the previous chapter yielded some limitations in handling unconstrained gaze estimation task. In such strategies, the head pose estimation and eye image analysis are processed and considered as separate blocks, each block produces a specific error giving a global accumulated gaze error. Nevertheless, performing a pose normalization in the second system achieves relatively good estimation results especially in strong head pose configurations.

In this chapter we present our third system defined as a fully appearance-based approach since the head pose and eye image analysis are treated implicitly as a single part. In the next sections, we give a global overview of this approach and discuss in details each part of the system. Our experiments demonstrate the potential of such paradigm in handling unconstrained gaze estimation ($i, e.,$: important head pose changes and large user-sensor distances). The obtained results in terms of accuracy

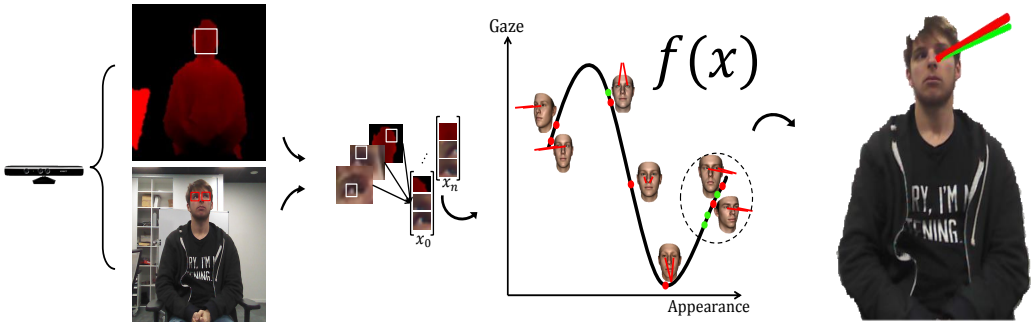
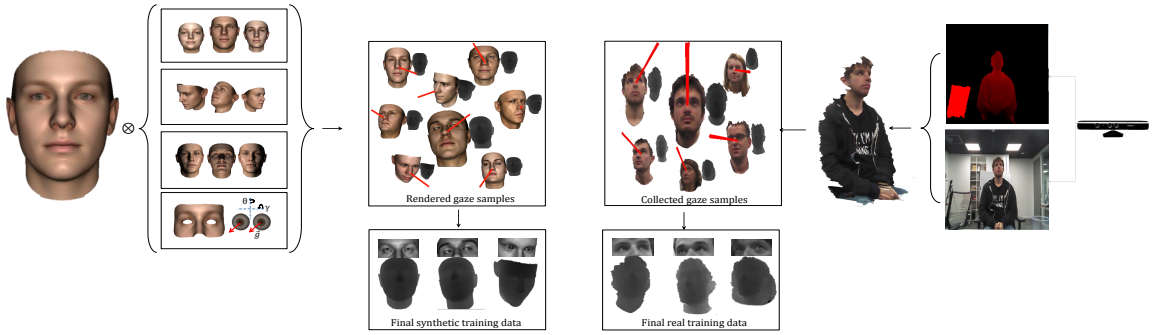


Figure 5.1: Example of automatic gaze estimation based on our approach. We build a 3-channels global vector represented by the two RGB eye images and the face depth information using the depth sensor multimodal data. We extract a set of patches and project it through the forest represented here as the mapping function $f(x)$ (the learned gaze sample clusters are defined as the red centroid points). Each single tree casts votes for each patch (defined as the green points). By performing a non-parametric clustering technique, a final estimation is calculated (represented as the green line, the red one defines the ground truth).

meet the technical requirements of our environment context discussed previously in chapter 1. We also perform an objective comparison with the semi appearance-based approach to establish a global conclusion through the different automatic gaze estimation strategies.

5.1 Overview

We consider the high non-linear problem of gaze estimation under head pose changes and large user-sensor distances as a regression task. To learn a robust high non-linear mapping function between human gaze and different gaze sample appearances, our fully appearance-based approach considers a global gaze manifold instead of learning in frontal configurations and geometrically correct the final estimation using head pose parameters, as usually is done. We train an ensemble of regression trees able to robustly capture gaze information on an important 3-channels training samples (channel$^{(0)}$:RGB-eye$_l$, channel$^{(1)}$:RGB-eye$_r$, channel$^{(2)}$:depth-face) organized as a set of patches (where a patch defines a small group of nearby pixels). We apply a channel-selection during the training to evaluate the importance and involvement of each channel in the final estimation. We define the gaze vector g as the vector stretching the gravity center of the face and the gazed 3D point. To provide a significant set of training data for the trees, we render a very important amount of gaze samples using a 3D statistical morphable model with integrating parametric gaze model. By parametric, we mean that our gaze model represented by two synthetic textured spheres, can be monitored yielding different gaze directions (we give more details in the next sections). We also build an important gaze database



(a) (b)

Figure 5.2: Data generation. (a) represents synthetic data rendering using the 3D morphable model of [Paysan 2009]. By introducing some variabilities such as identity, head pose changes and lightning conditions, we integrate a parametric gaze model (represented by two global textured spheres). We rendered the final RGB-D gaze training samples with the correspondent gaze annotation illustrated in red line. (b) We perform the same strategy using real data grabbed from the multimodal Kinect sensor by introducing the same previous variabilities. To obtain gaze annotation, a 2D moving point is gazed by the user (knowing the rigid transformation sensor-screen, the stretching vector from user head gravity center and the projection of the moving point in the world coordinates can be calculated). These real data are principally used as testing samples to evaluate accurately our forest trained on synthetic training set.

recorded with the Microsoft Kinect sensor. Rendered synthetic data and real data are used for both learning and testing, depending on the experiment.

Fig.5.1 describes an overview of our fully appearance automatic gaze estimation system that we define as S_2. We extract multi-channel patches from the multi-modal Kinect data, each patch is projected through a learned mapping function which gives a final estimation corresponding to the input. The mapping function, which corresponds to an ensemble of tree predictors $f(x)$ illustrated in Fig.5.1, represents the most significant part of the system. The estimation accuracy is directly linked to the ability of generalization of the learned trees. Providing representative training set in terms of quality and quantity is primordial. Fig.5.2 describes our generated training data used for the training.

5.2 Data generation

As noticed previously, the quality of the training set is primordial to ensure a sufficient gaze estimation accuracy and generalization of our regression trees. In this section we describe, as a first step, how we render synthetic gaze samples using morphable model to obtain a huge training set. Then, we detail how we recorded real

gaze training samples using the Kinect sensor. We describe the followed protocol to obtain ground truth information with a high reliability. In Fig.5.2, we present a global overview of the synthetic and real RGB-D gaze samples used as input to train our fully appearance-based automatic gaze estimation.

5.2.1 Synthetic data

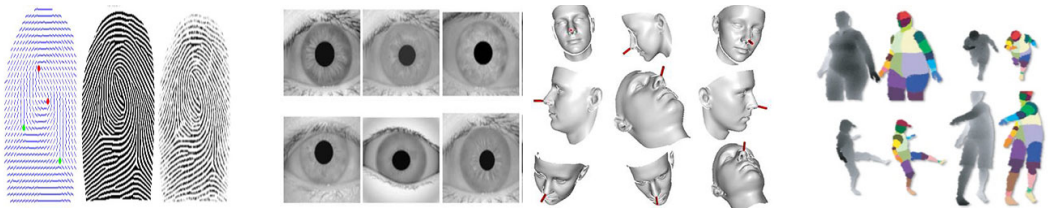
To highlight the importance of using synthetic data in computer vision, we give a short survey of some relevant works based on this kind of data, then we detail our synthetic gaze samples based on a face 3DMM with an integrated dynamic gaze model.

5.2.1.1 Synthetic data in computer vision

This last decade, machine learning techniques are considered as a very elegant way to tackle many problems in computer vision. Fig.5.3 illustrates some relevant works based on synthetic data. These data demonstrated a great potential in terms of efficiency and robustness. Nevertheless to achieve a high generalization across unseen scenarios, these methods often require a very representative training data set. But, the building of high amount of labeled data is a very tedious process. So synthetic data represents a promising solution as the annotation is performed automatically instead of manual labeling. [Cappelli 2000] developed an iterative model based on Gabor-filters applied on an empty image containing some seed points to render fingerprint training samples. [Zuo 2007] rendered iris image samples obtained from a 2D polar projection of a cylindrical representation of continuous fibers. [Thian 2003] improved face authentication by generating multiple virtual images using simple geometric transformations. [Shotton 2013] used a motion capture strategy to record RGB and depth cues of the body part movements, by varying body size and shape, scene position, camera position and mirroring the recorded data. They synthesize a highly varied training allowing a robust body part pose estimation. [Fanelli 2011] tackled the head pose estimation problem with synthetic depth images by rendering an important amount of training data using a 3D statistical morphable model (3DMM).

5.2.1.2 Synthetic data for gaze estimation

In our method, we first generate our synthetic training gaze samples by rendering a face 3DMM proposed by [Paysan 2009] which is the same model used for head pose estimation in [Fanelli 2011]. Concretely, this 3D morphable model consists in a set of texture and shape variation modes constructed from around 200 scans of human faces. These modes of variation represent an orthogonal basis able to model different face identities according to a linear interpolation. In Fig.5.4, we illustrate the deformations according to the texture and shape respectively starting from a mean configuration. To generate different face appearances, the model is deformed according to both directions.



(a) (b) (c) (d)

Figure 5.3: Some illustrative examples of using synthetic samples in computer vision. (a) fingerprint sample generation. (b) iris sample generation. (c) depth head pose sample generation (red line represents the head orientation). (d) body part sample generation (each part is represented as a single class with a specific color).

To build a such model, [Paysan 2009] performed 3 step as follows:

- 3D scans collection: a total of 200 face scans are captured (100 female and 100 male persons mostly Europeans, with ages between 8 and 62 years with an average of 25 years). To capture the face intrinsic characteristics, a structured light system is used including two projectors and 3 cameras producing 4 depth images. To achieve homogeneous illumination, 3 high fidelity cameras are used. Each scan is then preprocessed to remove air occlusions and different extraneous. In Fig.5.5a we shows un example of a raw and preprocessed scan used to train the 3DMM.

- 3D registration: the operation of registration means that all the preprocessed scans are re-parametrized such that corresponding points share the same position (for instance, the nose tips of two different scans share the same spatial information). To establish such correspondences, a non-rigid ICP algorithm is used which fills holes in missing regions by using robust distance measure. In addition, some landmarks (lips, eyebrows..) are manually added to improve the precision of registration. Fig.5.5b. illustrates some scans after performing the registration.

- Model training: After the parametrization, all the scans share the same topology. A face sample can be represented with two $3m$ vectors as follows:

$$sh = (x_1, y_1, z_1, x_2, y_2, z_2, ..., x_m, y_m, z_m)^\top \tag{5.1}$$

$$tex = (r_1, g_1, b_1, r_2, g_2, b_2, ..., r_m, g_m, b_m)^\top \tag{5.2}$$

where each vertex $(x_k, y_k, z_k)^\top \in \mathbb{R}^3$ with the associated color $(r_k, g_k, b_k)^\top \in [0\ 1]^3$, and a total number of vertices $m = 53490$.

Assuming independence between shape and texture, a Principle Component Analysis (PCA) is performed to the data yielding two linear models. The parametrization can be expressed as follows:

$$\mathcal{M}_{sh} = (\mu_{sh}, \sigma_{sh}, U_{sh}) \qquad \mathcal{M}_{tex} = (\mu_{tex}, \sigma_{tex}, U_{tex}) \tag{5.3}$$

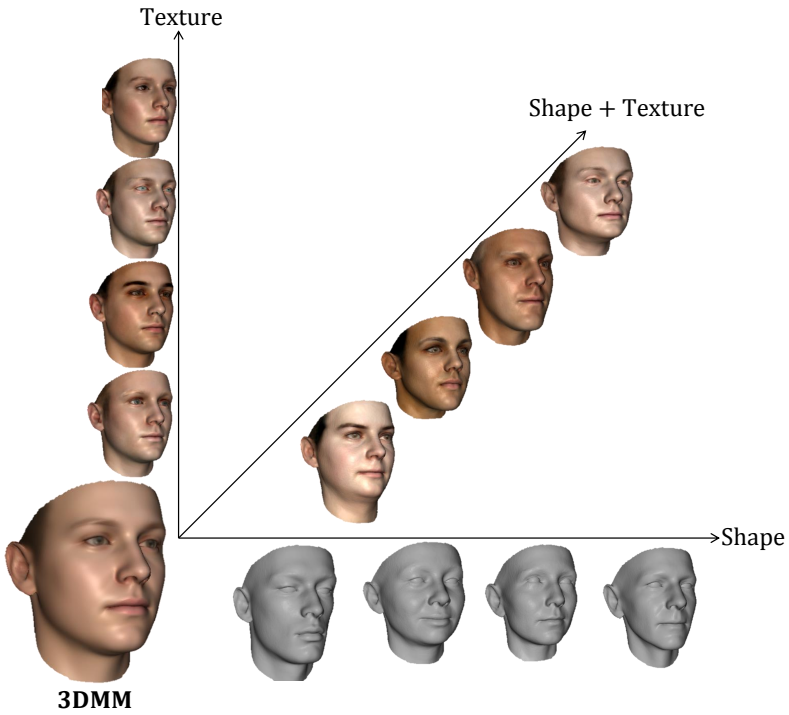


Figure 5.4: 3D morphable model of faces developed in [Paysan 2009]. The model is deformed from the mean configuration (in terms of shape and texture) according to two directions, texture and shape respectively. The texture dimension changes the RGB information of each vertex v belonging to the model, whereas, the shape changes the spatial information (x, y, z) of each vertex. Deforming the model in both directions yields different face identities.

where $\mu_{j\in\{sh,tex\}} \in \mathbb{R}^{3m\times 1}$ encodes the texture and shape mean respectively, $\sigma_{j\in\{sh,tex\}} \in \mathbb{R}^{n-1} \times 1$ the standard deviation and $U_{j\in\{sh,tex\}} \in \mathbb{R}^{3m\times n-1}$ are the orthogonal basis encoding the variation modes of the model. To generate a random face, a linear combination of the principle components is applied as follows:

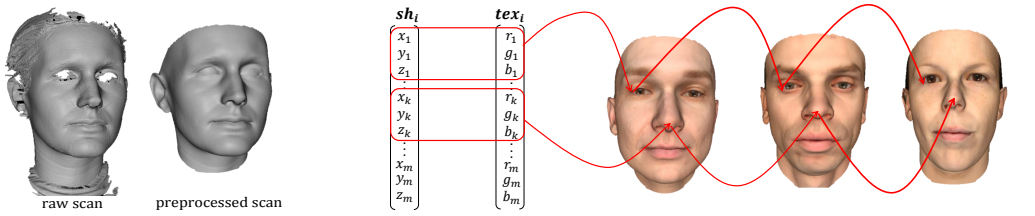
$$sh(\alpha) = \mu_{sh} + U_{sh}.diag(\sigma_{sh}).\alpha \tag{5.4}$$

$$tex(\beta) = \mu_{tex} + U_{tex}.diag(\sigma_{tex}).\beta \tag{5.5}$$

where $\alpha, \beta \in \mathbb{R}^{n-1\times 1}$ are random coefficients.

Unfortunately, there are no principle components (in both texture and shape information) responsible of gaze variability. Since the scans are captured with natural configuration under no eyes movements, the PCA encodes no information about gaze.

As can be seen, the 3D face morphable model procedure is a very tedious task which can take a very long time to obtain sufficient precision and good quality. Instead of building a new model, we decided to integrate to the 3DMM (described



(a) (b)

Figure 5.5: 3D morphable model, acquisition and registration. (a) describes the original scan and the preprocessing one used to train the model. (b) illustrates examples of face scan after registration, the texture and shape information is organized as vectors tex and sh respectively (represented as one column vector with a dimension $3m, m$ is the total number of vertices). The same points in the parametrization domain (sharing the same index), extracted from different scans, correspond to the same semantic regions.

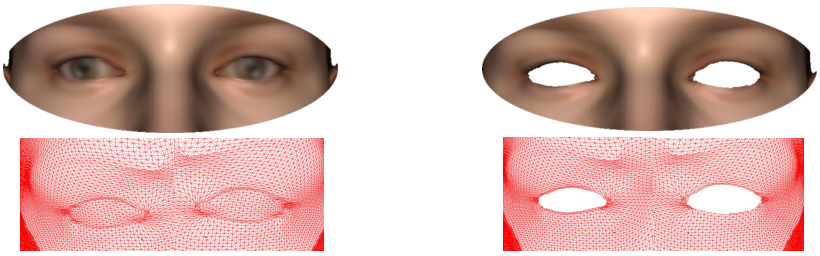


Figure 5.6: Eyes vertices removing. On the right, the original model and on the left, the model with empty eyes region. The red images describes triangulation of the model.

above) a dynamic gaze model able to generate different gaze samples (which can be considered as an extension of the basis component) and to maintain the same parametrization and topology.

We integrate our gaze model in 3 steps as follows:

- Eyes vertices removing: we identify manually all the vertices inside eye regions (580 vertices identified), and remove all the triangles which include at least one vertice in its edges (we reduce the number of triangles from 106466 to 105151). Fig.5.6 illustrates the model before and after removing eyes regions.
- Eyelids managing: to control the eyelids movements resulting from an upward and downward gazing, we introduce a linear translation for each vertex surrounding the eye regions. We express the localization of these vertices as a function of latent variable ε as follows:

$$v_i(\varepsilon) = a_i + b_i\varepsilon \tag{5.6}$$

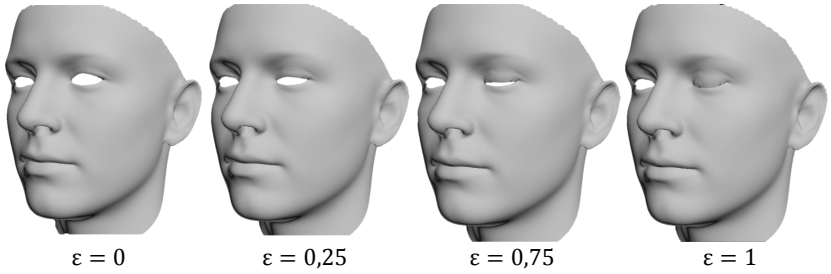


Figure 5.7: Eye lids movements managing. By selecting a set of points around the eyes, we define blendshapes able to monitor eyes closure through a global parameter ε

with $\varepsilon \in [0\ 1]$, ε =0 corresponds to the starting position of the vertices and $\varepsilon = 1$ to the final position which are manually fixed (gazing down and up are treated as separate configurations with different starting and final localization). All the coefficients of the linear translations (a_i, b_i) can be expressed as $(v_i(0), v_i(1) - v_i(0))$. Thanks to the topology of the model, all these modifications keep the same behavior under identity variation. Fig.5.7 shows some results of eyelids movements according to the defined linear parametrization.

- Eyeballs setting: we place two spheres as eyeballs. We fix the diameter to the human average eyeball namely 25 mm. We use different textures for the eyeballs to handle the iris appearance variability. To generate gaze sample, we generate a virtual 3D point on which the two eyeballs turn toward, the gaze information angles can be easily calculated knowing the location of the eyeballs centers. To integrate the eyelids movements in the gaze samples, we establish a linear correlation between ε and the gaze vertical angle γ for downward and upward gazing separately.

Fig.5.8 illustrates some examples of the final RGB-D gaze samples used to train our system.

5.2.2 Real data

Our major motivation for building real gaze RGB-D samples is to evaluate the synthetic training data through different objectives comparisons. Our experiments allow us to define an existing empiric relation between synthetic and real data in terms of quality and quantity in handling the task in hand.

To record real RGB-D gaze samples, we use a Kinect sensor with (1280×960) and (320×240) resolutions, for RGB and depth streams respectively (the depth map is re-sampled to the same resolution as the RGB to establish the mapping between the two maps). Our database contains 42 peoples, 15 females and 27 males, 4 wearing glasses and 38 without glasses. A total of 17k RGB-D samples is recorded.

The participants are asked to gaze a moving 2D points $m(u, v)$ (expressed in the image space) displayed on a planar screen under 8 configurations as follows:

Figure 5.8: Some gaze sample examples. Notice the eyelids behavior according to gaze direction especially to the last three faces.

- Seated-frontal: the user gazes different points m_i with no head movements. The distance user-sensor is fixed to 150 cm, a total of 30 RGB-D samples per user is recorded.
- Seated-head pose changes: under the same distance as the previous configuration, the user gazes the point with performing a continuous head pose movement at the same time. The user remains seated, a total of 70 samples per user is recorded.
- Standing-frontal: the same conditions as in the first configuration are applied except that the user is standing. A total of 40 samples is recorded.
- Standing-head pose changes: the user performs the same instructions as in configuration two in a standing way. A total of 80 samples is recorded.

The four last configurations are similar to the previous ones under a user-sensor distance of 200cm. The number of samples per configuration and per user are kept fix. In Fig.5.9, we show for a given participant, the RGB-D gaze samples result under the four first configurations.

As we can see in Fig.5.9, the final gaze ground truth is represented as a red cylinder stretching the face gravity center and the 3D projection $M(x_M, y_M, z_M)$ of $m(u_m, v_m)$. To perform the projection of the target point m into the Kinect coordinate system, we initially project m to M' using the pinhole projection as illustrated in the equation. 5.7. with :

$$\begin{cases} x_{M'} = \dfrac{d_m.(u_m - c_x)}{f_x} \\ y_{M'} = \dfrac{d_m.(v_m - c_y)}{f_y} \\ z_{M'} = d_m \end{cases} \tag{5.7}$$

where d_m defines correspondent depth value of m.

Knowing the rigid transformation $[R'|T']$ between the planar screen (with a dimensions of 187 cm and 105cm in width and height respectively) and the Kinect

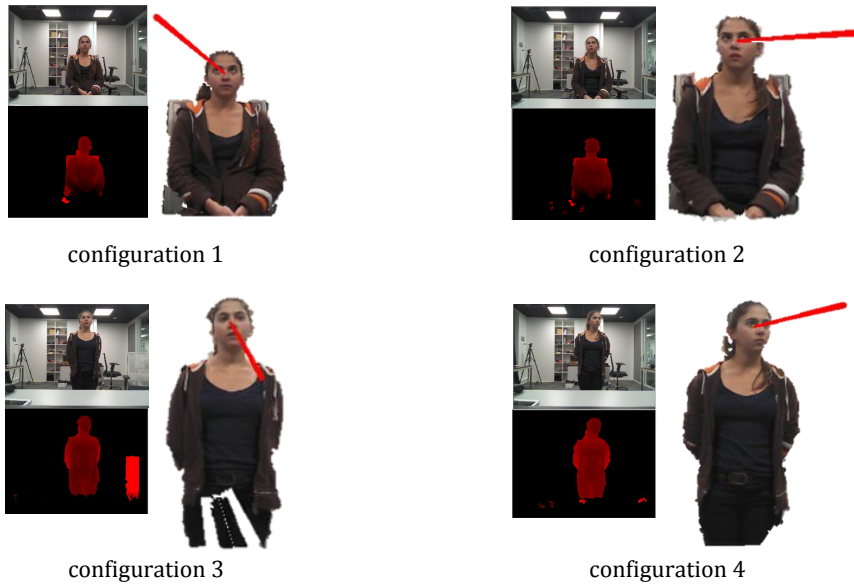


Figure 5.9: Examples of our RGB-D gaze samples under different configurations using Kinect sensor. Using RGB-D stream, a textured mesh is constructed, the red cylinder describes the gaze information ground truth.

sensor, the point M' can be transformed to M as follows:

$$\begin{bmatrix} x_M \\ y_M \\ z_M \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} \times \begin{bmatrix} x_{M'} \\ y_{M'} \\ z_{M'} \end{bmatrix}^{\top} + \begin{bmatrix} T'_x \\ T'_y \\ T'_z \end{bmatrix} \tag{5.8}$$

In our case, we setup the Kinect to approximate the maximum R' as the identity matrix I. The translation matrix T' is estimated to $(93.3, 99.1, -4.0)$. In Fig.5.10 we show the experiments setup. We illustrate the position of the sensor (in green) according to the screen (in red). The user is gazing a target point m initially expressed in the image space then projected into the Kinect space (represented in white dotted lines) giving the gaze information. Fig.5.10 also describes some final RGB-D gaze samples relative to some participants, including the gaze ground truth (green cylinders).

To express the gaze information g as two angles (θ, γ) as performed in the previous chapter, we define a reference point $O(x_O, y_O, z_O)$ as the displacement of the face gravity center G along the z *axis*. We finally compute the angles between (θ, γ) $\overrightarrow{GM}$ and $\overrightarrow{GO}$.

5.3 System training

We train different gaze estimation forests $\mathcal{T}_g$ depending on the nature and the number of training data used. Each tree χ in each forest is trained in a supervised way as

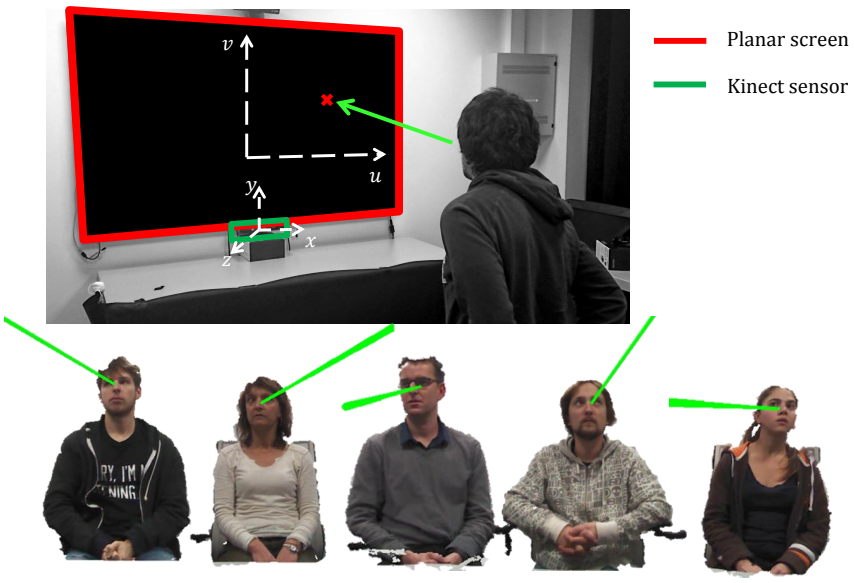


Figure 5.10: Establishing gaze ground truth. Knowing the screen-sensor rigid transformation, each gazed target can be projected in the world coordinate system using pinhole model. The final gaze vector is defined as the vector stretching the user head gravity center and the 3D gazed point (illustrated with the green cylinder).

described previously. We provide for each a significant training set $\{\mathcal{P}_i = (\mathcal{I}_i^o, y_i)\}$ with:

- $\mathcal{I}_i^c$ the extracted visual features vector contains 3 channels ($o = \{0, 1, 2\}$).The channels correspond to the two RGB intensities extracted from the two eye images and the depth values extracted from the face.
- g_i represents the output gaze vector represented with two components (θ, γ).

The face depth image size is fixed to (150×150) (the red face rectangle represents the output of the Viola Jones face detection performed on the RGB image represented in Fig.5.11). The two eye images size is fixed to (80×70) (we use the same anthropomorphic relations as performed in the pupil localization described in the previous chapter). Each channel $\mathcal{I}^o$ size is fixed to (16×16). Fig.5.11 illustrates how we perform multi-channel patches extraction as input to train our trees.

We describes the different forests $\mathcal{T}_g$ trained as follows:

- $\mathcal{T}_{g_0}$ represents our principal gaze estimation forest. We provide 400k synthetic training RGB-D gaze samples under strong head pose changes and scale variation. We extract 15 3-channels from each sample giving a total of 6M training patches. Each tree predictor is trained on 60% of the global set giving 3.6M training data. Some training parameters are fixed to some empirical observations, *e.g.*, the stopping criteria are fixed to 18 and 100 for maximum depth

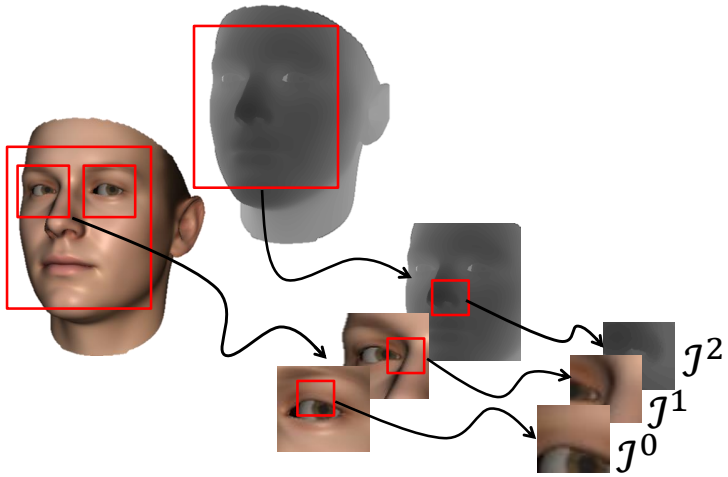


Figure 5.11: 3-channel patch extraction example. Face detection is performed on the RGB image which is mapped on the depth image. Using geometric assumptions, we extract two rough regions around the eyes, a multi channel input is then built including two RGB eyes and depth face. From each region, we randomly extract a fixed size patch producing the final 3-channel patch $\mathcal{I}^o$.

tree and minimum samples at leaves respectively. To optimize the splitting at each node, we generate a pool of 400 candidates with 50 threshold values giving 20k binary tests. $\mathcal{T}_{g_0}$ contains a total of 40 trees.

- $\mathcal{T}_{g_1}$ is a gaze estimation forest trained on synthetic patches extracted from RGB-D in exclusively frontal configuration with scale variation. Each tree is trained in the same way as performed in $\mathcal{T}_{g_0}$ with exactly the same number of training data.

- $\mathcal{T}_{g_2}$ represents a gaze estimation forest trained exclusively on real RGB-D gaze samples described previously. Our main motivation of training this forest is to perform an objective comparison with a forest trained on synthetic data in terms of gaze estimation accuracy. We provided for each tree $200k$ training patches extracted from a global set of $500k$. We generate $1k$ binary tests at each node which is considerably low comparing to $\mathcal{T}_{g_0}$ and affects directly estimation accuracy. The principal reason of alleviating the node optimization is to build different trees with synthetic data able to be compared in reasonable time. The stopping criteria is kept fix.

- $\mathcal{T}_{g_s}$ represents different forests trained on different number of synthetic data namely $200k$, $500k$, 1M and 5M (the RGB-D sample are under head pose changes and scale variation). Each predictor from each forest takes 60% from the correspondent training set and is trained in the same way as $\mathcal{T}_{g_0}$.

In addition to these gaze estimation forests, we trained another head pose forest $\mathcal{T}_{g_h}$ following the same learning strategy discussed in the previous chapter. We

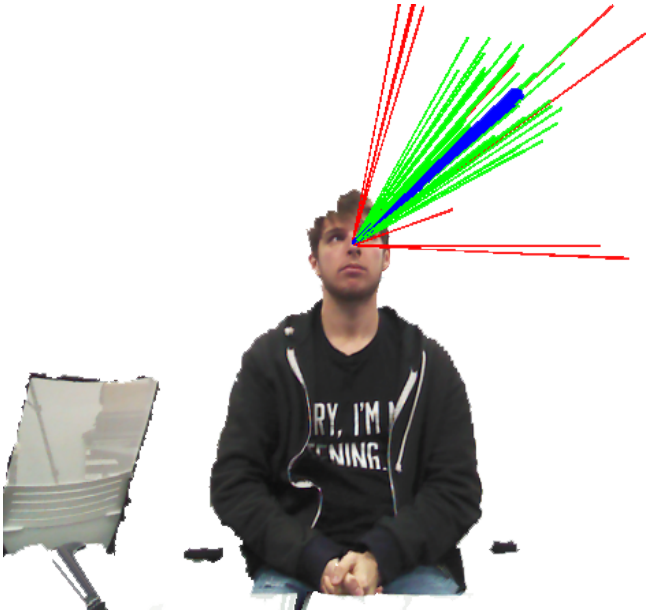


Figure 5.12: Test instance example: the votes selected by mean-shift filtering are represented in green, the non-informative leaves responses are in red and the final estimation is in blue.

provided a training set of $50k$ synthetic training RGB-D head samples with $20k$ binary tests. We fixed the depth tree to 15 with a minimum samples to 50.

5.4 Experiments

To estimate the gaze vector from an unseen instance, we extract a set of patches from the RGB eye regions and the face depth information after a face detection step. Each patch is passed through all the learned trees in the forest using the optimal stored binary tests as described previously.

All the estimations corresponding to the extracted patches are regrouped in votes. Before performing the clustering of these votes, we discard the estimations from the leaves with high variance considered as non-informative. To locate the centroid of the cluster of the votes, we perform 5 mean-shift iterations using a Gaussian kernel. Fig.5.12 shows an example of the final estimation, the green ones represent the votes casted by the forest which are selected by the mean-shift. The red lines correspond to some casted votes with a high variance discarded by the mean-shift. The final estimation is given by the blue line corresponding to the centroid of the selected votes.

We perform experiments with different scenarios. First we evaluate the gaze accuracy as a function of the testing forest parameters (number of patches and trees), then we quantify the precision of our main forest and its ability to handle real unconstrained configurations (even if the predictors are exclusively trained on

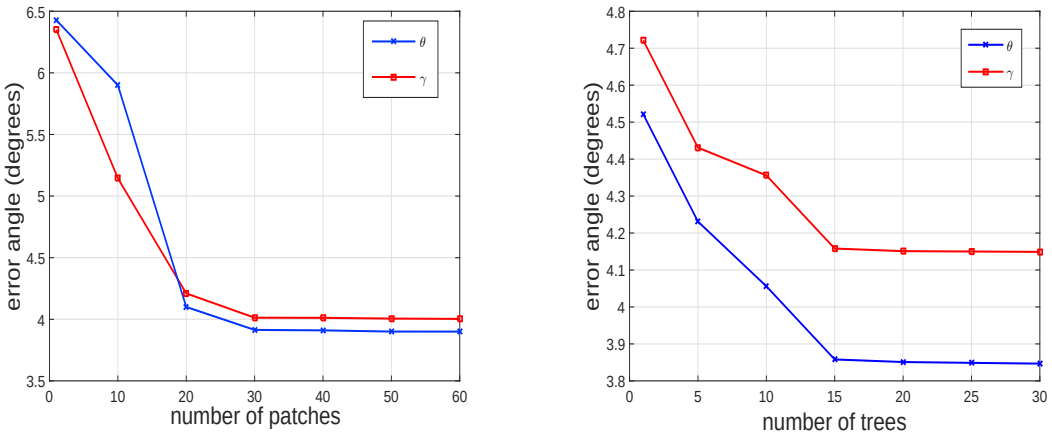


Figure 5.13: Forest parameters effect on the gaze estimation accuracy. (a) describes the behavior of the gaze mean error over 25 users (according to the two directions θ and γ respectively) as a function of the number of patches extracted as input for the forest. (b) illustrates the same errors by varying the total number of trees included in the final decision casted by the forest.

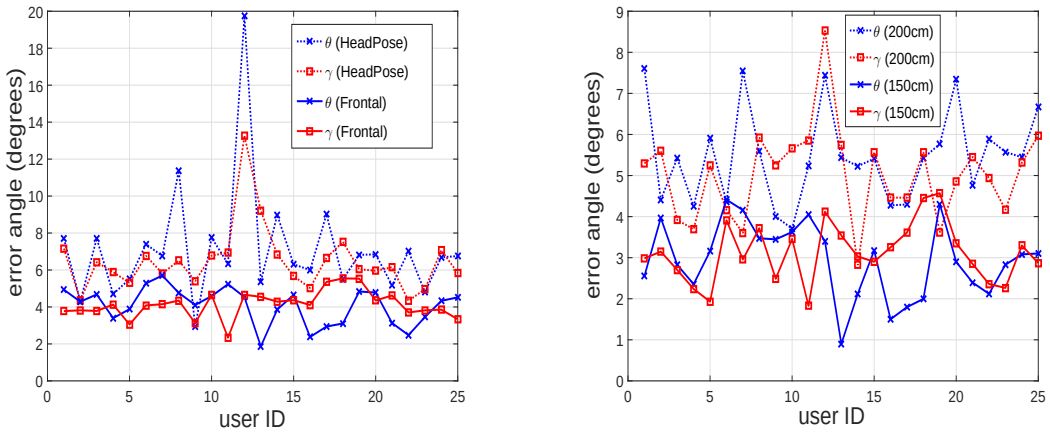
synthetic data). We also evaluate the involvement of each channel and provide concrete results of the importance of the depth information. We finally achieve an objective comparison on synthetic-real learning.

5.4.1 Forest decision parameters effect

We use the forest $\mathcal{T}_{g_0}$ to determine the optimal testing forest parameters namely the number of extracted patches and the forest size. We perform our tests over 25 participants provided with the gaze ground truth as explained in the data generation section. The user-sensor distance is fixed to 150cm with low head pose variations. Fig.5.13 summarizes the obtained results of our experiments.

Fig.5.13a illustrates the gaze mean error (through θ and γ) over all the participants under different number of extracted patches values. The error decreases by increasing the numbers of patches. These errors are reduced approximatively by 40% (from 6.5° to 3.9° for θ, and 6.5° to 4.0° for γ). This result is expected since the forest captures more information about input when increasing the number patches as noticed for the pupil localization. Increasing the value over 30 has insignificant effect on the estimation, so we decided to fix the number of patches for testing for the next experiments to this value.

Fig.5.13b describes the behavior of the mean gaze error as a function of the number of trees included in the final decision forest $\mathcal{T}_{g_2}$ (the number of the extracted patches is fixed to 30). The error decreases by approximatively 12% (from 4.5° to 3.8° for θ, and 4.7° to 4.1° for γ). As shown in the pupil localization forest analysis,



(a) (b)

Figure 5.14: Evaluation of the gaze estimation accuracy under unconstrained scenarios using using $\mathcal{T}_{g_0}$. (a) the mean error for the two gaze directions under frontal and head pose changes. (b) the mean error for the two gaze directions under two distances from the sensor.

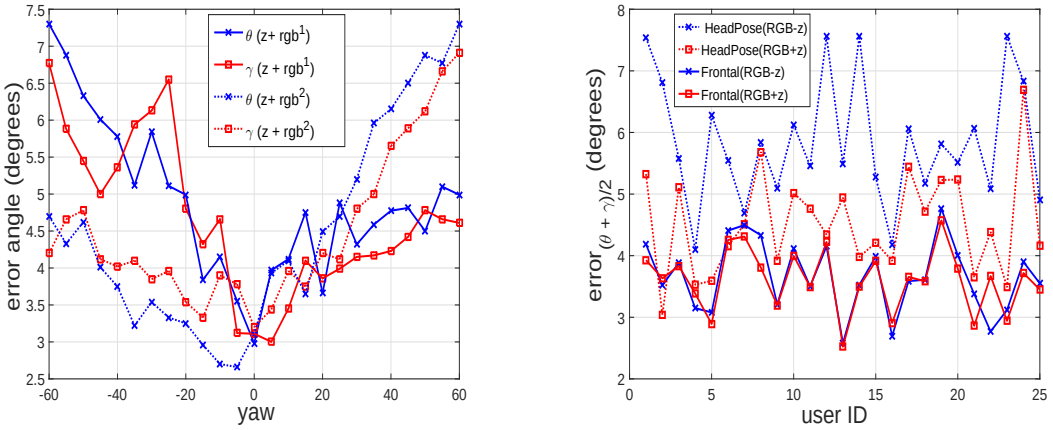
increasing the forest size performs more generalization hence produces more robust and accurate output. We fixed the optimal number of trees to 15.

5.4.2 Robustness to head pose and distance variation

We evaluate the gaze estimation accuracy using our trained forest $\mathcal{T}_{g_0}$ under unconstrained environment. Our estimation is based on the forest $\mathcal{T}_{g_0}$ with 30 and 15 as number of extracted patches and forest size respectively. In these experiments we show the influence of head pose movements and the user-sensor distance on the estimation accuracy. Fig.5.14 resumes the obtained results.

Fig.5.14a represents the global error of gaze estimation over 25 users under frontal and head changes configurations. For each user, a mean error across different gaze samples performed under two distances is computed. In frontal case, the mean error over all the users is less than 3° for the two directions whereas the error is less than 6.5° for head pose changes case. This difference in accuracy between the two configurations is directly linked to the high eye image appearance variability across head pose configurations, making the trees prediction less accurate.

In Fig.5.14b we report the error as a function of distance from the sensor for a frontal configuration. The experiments show a mean error of 2.9° and 3.1° for θ and γ respectively at 150 cm from the sensor. At 200cm, we notified slightly higher errors, 4.8° and 5.0° for the two directions respectively. The difference in accuracy between the two distances is related to the RGB eye images and face depth appearance which are significantly variable depending on the distance to the sensor.



(a) (b)

Figure 5.15: The involvement of each channel in the gaze estimation decision casted by the forest. (a) mean error of the two directions over head pose variations (yaw angle variation) with different channels combinations. (b) mean error over the two directions with and without using depth information in frontal and head pose configurations respectively.

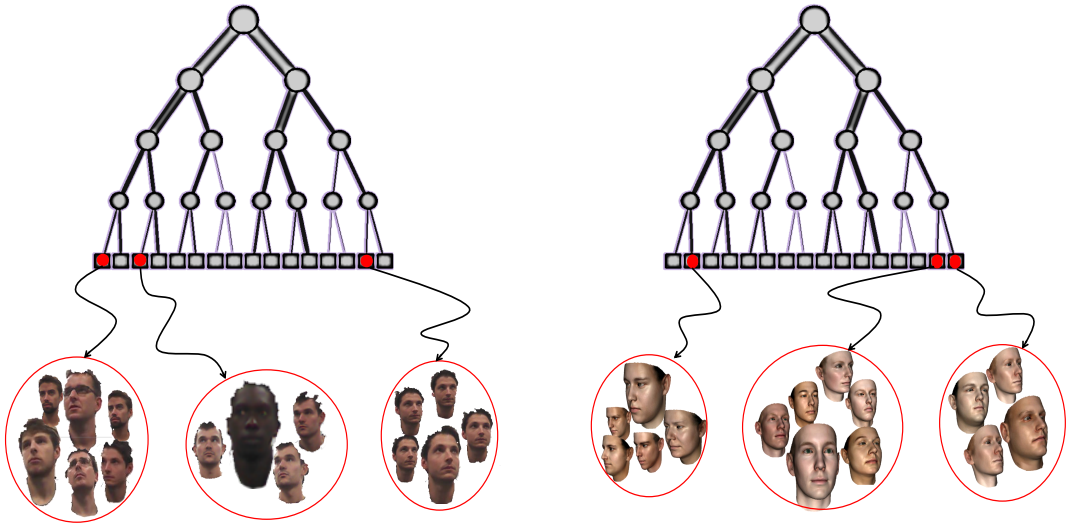
5.4.3 Channel selection importance

To evaluate the involvement of each channel (from the two eye RGB images and face depth information) in our fully appearance based automatic gaze estimation system, we perform experiments using a specific trained forest $\mathcal{T}^*_{i\in\{0,1,2\}}$ as follows:

- $\mathcal{T}^*_0$: trained as $\mathcal{T}_{g_0}$ on a set of patches with only the two RGB channels and no depth information.
- $\mathcal{T}^*_1$: trained on a set of patches with two channels, depth information and RGB extracted from the left eye.
- $\mathcal{T}^*_2$: trained on a set of patches with two channels, depth information and RGB extracted from the right eye.

Fig.5.15 reports the obtained results on gaze estimation accuracy using these specific forests.

Fig.5.15a describes the influence of the two RGB channels (corresponding to right and left eyes) on gaze estimation accuracy across different yaw angle values. We evaluate the accuracy using $\mathcal{T}^*_1$ and $\mathcal{T}^*_2$ respectively. These results are expected since eye appearance is very sensitive to head pose changes especially for *yaw* angle variation. For instance, positive values of yaw deform the left eye appearance until a complete disappearance giving high estimation errors for the two directions without the visible channel namely right eye ($i, e.$, dotted lines in Fig5.15a) and reciprocally. Using the channel selection strategy introduced on the forest learning,



(a) (b)

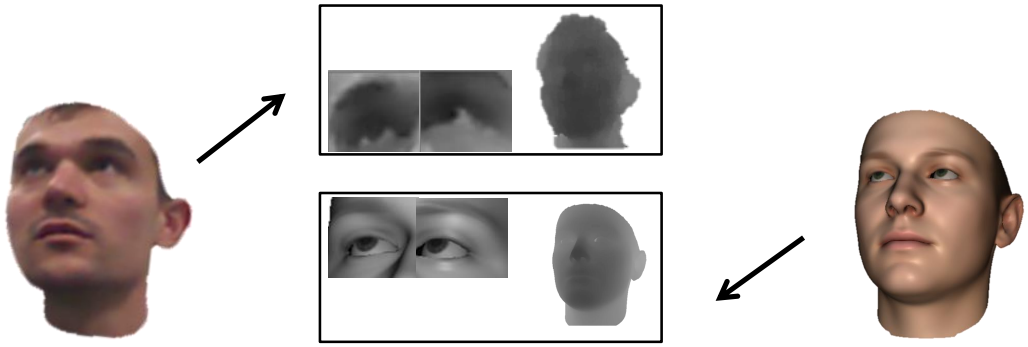
Figure 5.16: Visualizing some clusters captured during the training step of $\mathcal{T}_{g_0}$ and $\mathcal{T}_{g_2}$ using only depth cue ($\mathcal{I}^2$). (a) describes the reached real data on 3 leaves (represented with red circles). (b) illustrates the same experience with synthetic samples.

we can quantify the involvement of each RGB channel in the final gaze estimation across head pose changes.

Fig.5.15b illustrates the importance of depth information in our approach especially in head pose changes scenarios. We test on previous participants gaze accuracy with $\mathcal{T}_0^*$ and $\mathcal{T}_{g_0}$ respectively. Gaze estimation errors are very close with and without depth information in frontal scenario whereas the error gap is approximatively 1.5° in head pose changes configuration proving the importance of this channel in such case. Depth information is more suitable to encode geometric similarities between data samples which represent the head pose information. In Fig.5.16, we show some clusters with low variances captured by the forest $\mathcal{T}_0^*$ using real and synthetic data respectively. As we can see in both cases, the data which reach the specified leaves (with red circles) present semantic information about the gaze (for each cluster, the participants look approximatively in the same direction). In reality, the data reaching leaves are patches. For this experiment each patch $\mathcal{P}_\rangle$ is represented as $(\mathcal{I}_i^o, g_i, x_i)$ where x_i is the original RGB-D sample where the patch is extracted. x_i allows us to illustrate the data for each cluster as RGB-D samples.

5.4.4 Learning with real data versus learning with synthetic data

In this section, we evaluate the realism of our rendered synthetic data and their ability to handle unconstrained gaze estimation problem. We perform experiments using the forest $\mathcal{T}_{g_2}$ trained on real data and forests $\mathcal{T}_{g_s}$ trained on different number



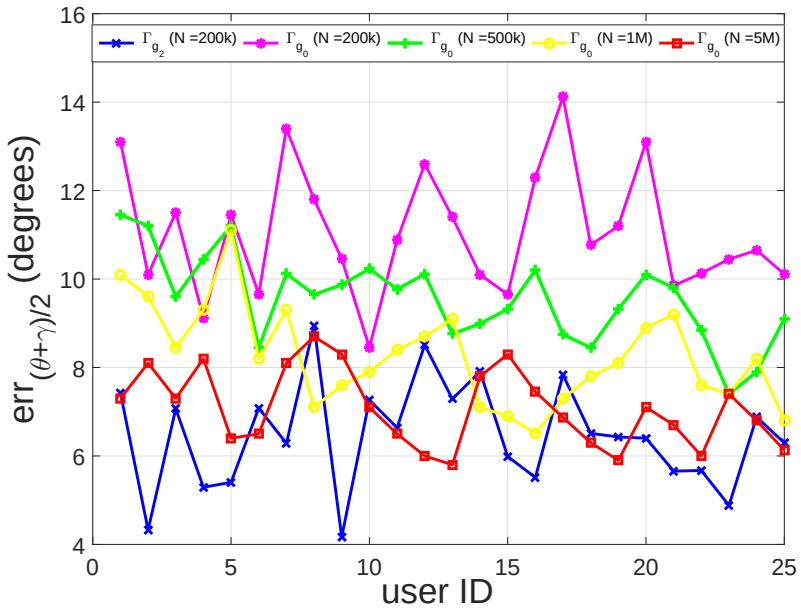
(a)

Figure 5.17: Visual comparison between real and synthetic RGB-D sample.

of synthetic training data. The forests are tested on the same participants described previously. The main motivation of these experiments is to quantify empirically the existing ratio between the numbers of real and synthetic training sets producing similar results.

In Fig.5.17 we show a visual comparison between real and synthetic RGB-D samples. The real sample is a testing participant extracted from our database described before, the synthetic one is generated using the 3DMM with our parametric-gaze model with the mean shape and texture. We choose two RGB-D corresponding approximatively to the same gaze direction which gives a first opinion about the similitude between the two data. The depth information looks globally similar in both cases except the existing noise on the real data (due to noisy depth imaging of the Kinect sensor). For the two eye regions, the similarity is more apparent on the patches near to the pupils (which represent the most important information about the gaze). The patches surrounding the pupils present differences which can be explained by the lightness of the 3DMM texture.

In Fig.5.18 we report quantitative results on the gaze estimation accuracy performed by $\mathcal{T}_{g_0}$ and $\mathcal{T}_{g_s}$. We average the gaze error over θ and γ and over the two distances 150cm and 200cm respectively. By using the same number of training data namely $200k$, $\mathcal{T}_{g_2}$ and $\mathcal{T}_{g_s}$ produce different error curves with an apparent gap (the mean errors are 6.4° and 11.1° for the real and synthetic forest respectively). This significant difference in accuracy ($\sim 4°$) is directly linked to the difference in the representativity of the training example. Since synthetic data aims to reproduce the same appearance as the real data but with some differences, their corresponding gaze manifold learned at the training step does not cover all the real scenarios. Unlike real samples which build a manifold more closer to the testing configuration. Increasing the training set provided for $\mathcal{T}_{g_s}$ decreases the gaze estimation error (9.5°, 8.3°, 7.1° for $500k$, 1M and 5M training data respectively). Increasing the cardinal of the training set enhances generalization ability of the learned tree predictors across unseen testing scenarios. The empiric ratio between synthetic and real data



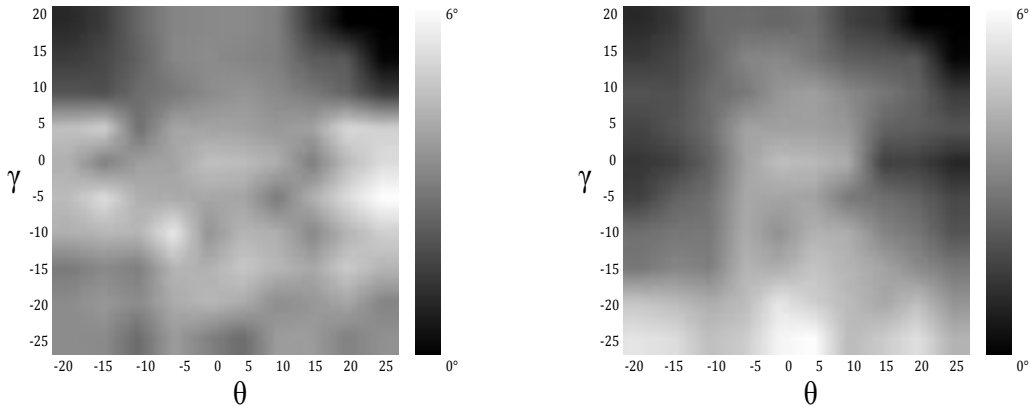
(a)

Figure 5.18: gaze estimation error (over the two directions and over the two distances 150 cm and 200 cm)under learning with real and synthetic data (N is the number of training data used)

producing approximatively the same performance in terms of accuracy is estimated to 1/9. This results presents an important meaning for our purpose. Since the generation of labeled synthetic RGB-D gaze samples is more effortless and faster with more accurate ground truth compared to the real samples, they can at the same time produce similar performance for gaze estimation by enhancing the training set following this empirical ratio.

In Fig.5.19 we report the gaze estimation error distribution across the 5 best testing participants using $\mathcal{T}_{g_2}$ and $\mathcal{T}_{g_0}$ respectively. In Fig.5.19a we describe globally a uniform distribution with high values for θ superior to 20° corresponding to the right gazing. This result explains the limitation of the generalization of forest $\mathcal{T}_{g_2}$ in handling some extreme unseen scenarios. The gaze error distribution is represented in Fig.5.19b.We can distinguish 3 regions as follows:

- $\gamma < -20°$ represents the highest error range. These γ values correspond to the eyes closure making the eye image appearance very similar even if θ is varying which produces bad gaze estimations. Furthermore, our dynamic gaze model performs a linear shifting on the eyelid vertices to cover the new eye shape and stretches the original eyelid texture to cover the new texture giving a rough approximation of the real eye appearance. Our choice of the dynamic gaze model is strongly constrained by the 3DMM topology.



(a) (b)

Figure 5.19: Mean gaze error distribution over 25 participants across different gaze directions ((a) real testing samples, (b) synthetic testing samples).

- $|\theta| < -7°$ describes a region with a relatively important error. Our forest is weakly discriminative with straight gazing samples under large distances. In addition, we noticed, for some users, an important error for upward gazing configuration($\gamma > 10°$ and $\theta < 5°$) which can be explained by an elliptical deformation of the high part of the eyes. In fact, this deformation is very person-specific and our dynamic model performs the same deformation over the different face shapes generated by the 3DMM, the forest gives less accurate results.
- $\gamma > -20°$ and $|\theta| > 5°$ cover the range of good gaze estimation (error less than 4°) which represents more than 50% of the total area. The appearance of the patches extracted from these gaze samples are very discriminative. In addition, for these configurations, our synthetic training data present a very high realism.

5.5 Fully appearance-based versus semi appearance-based approach

To achieve a comprehensive comparison between fully and semi appearance-based approaches in handling of unconstrained gaze estimation, we perform experiments over the 25 synthetic users under 3 scenarios:

- Estimation performed using the forest $\mathcal{T}_{g_0}$ under a user-sensor distance of 150cm.
- Estimation performed using $\mathcal{T}_{g_1}$ (trained only on frontal appearances), using the head-sensor rigid transformation, the final gaze estimation is obtained as

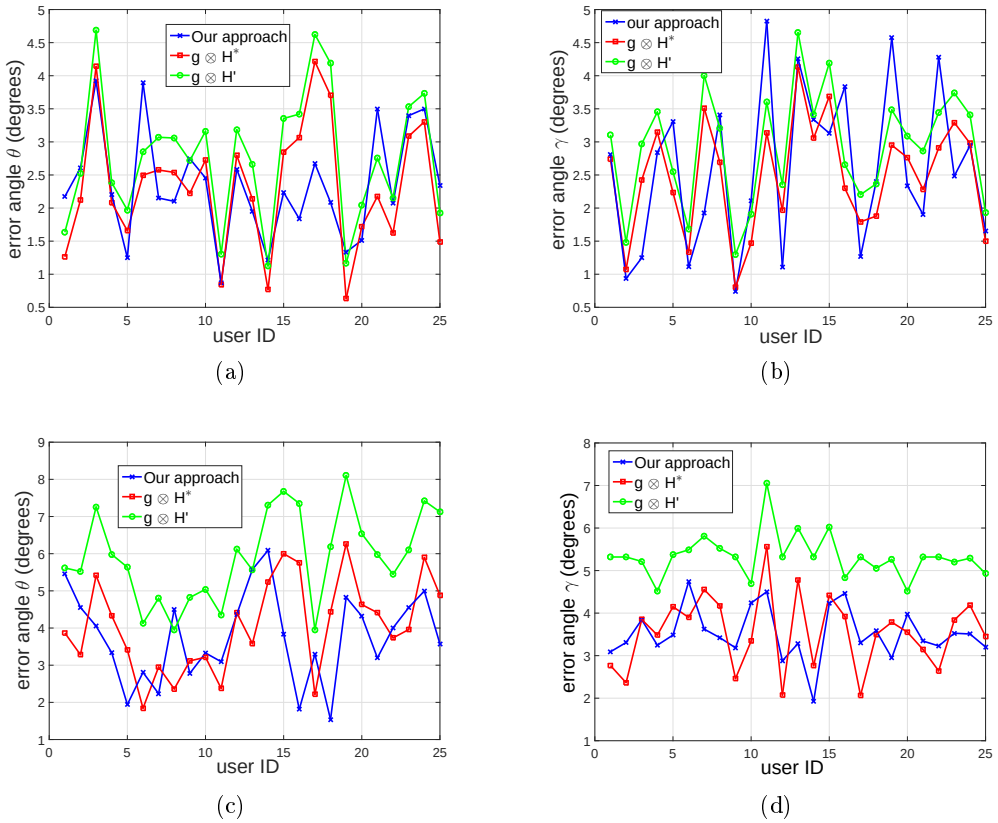


Figure 5.20: Comparison of our approach to semi appearance-based approach using exclusively synthetic data. Our approach (in red) performs gaze estimation using RGB-D cues assuming a global gaze manifold(under head pose changes) in red. Semi appearance-based approach (in green) performs gaze estimation assuming frontal configuration with a geometrical correction using estimated head pose parameters. In blue: a semi appearance-based approach with ideal head pose parameters. (a) and (b): gaze errors in frontal configuration for θ and γ respectively. (c) and (d): gaze errors in head pose change configurations.

explained in the previous chapter. The head pose parameters are directly driven from the OpenGL camera model which is considered as a ground truth value.

- Estimation is performed following the same strategy as in the scenario 2 except that in this case, the head pose parameters are estimated using the forest $\mathcal{T}_{g_h}$.

The main objective of this experiment is to provide an empirical proof of the advantages of the fully appearance-based approaches. We report the obtained results in Fig.5.20.

Fig.5.20 illustrates the estimation errors under the three described scenarios under frontal configurations through θ and γ respectively. In this figure we represent

the semi-appearance based system using $g \otimes H^*$ and $g \otimes H'$, this notation describes the correction of the frontal gaze g using ideal head pose H^* and estimated head pose H' respectively. The three scenarios present mean errors of 2.4°, 2.3° and 2.7° for θ and 2.6°, 2.4° and 3.0° for γ (these errors are sensibly low which is the result of using synthetic instead of real samples for the testing, and the forests used are trained on the same nature of data). The errors are very close which can be explained by the fact that the different systems perform the same behavior under frontal configuration. For $\mathcal{T}_{g_0}$, the testing samples reach the leaves with frontal appearance samples where the depth information is slightly used during the training. The head pose parameters are estimated with a sufficient accuracy by $\mathcal{T}_{g_h}$ in frontal appearance for the third system (and absolutely correct for the second one), giving an accurate geometric correction for the final estimation.

Fig.5.20c and 5.20d describe the behavior of the previous systems using the same testing samples generated under strong head changes. We noticed gaze estimation errors of 3.8°, 4.0° and 5.9° for θ, and 3.5°, 3.5° and 5.3° for γ. The fully appearance-based approach presents the lowest errors through the two gaze directions. Gaze estimation in head pose changes involves strongly head pose cues, thanks to the depth information discrimination, $\mathcal{T}_{g_0}$ achieves good estimation even with head pose changes as illustrated in the channel selection importance section. Correcting gaze with head pose parameters produces a cumulative error related to the frontal estimation and head-sensor rigid transformation respectively. For strong head appearances, the third system performs a head pose estimation with an important making a high final gaze estimation. This experiment gives a strong empirical proof of an optimal automatic gaze estimation system, according to our results and our environmental conditions. Considering head pose and gaze information as two independent blocks produces important errors for relatively high user-sensor distances.

Methods	75cm	150cm	200cm
[Chen 2008](F)	2.35°(< 75cm)	-	-
[Jianfeng 2014](F)	4.8°	-	-
[Mora 2012](H)	7.1°	-	-
Our feature-based(F)	3.8°	5.1°	-
Our feature-based(H)	7.8°	9.8°	-
Our Semi.App-based(F)	7.0°	7.3°	-
Our Semi.App-based(H)	-	8.1°	11.3°
Our Fully.App-based(H)	6.1°	7.1°	7.6°

Table 5.1: Comparison of our automatic gaze estimation systems with the state-of-art methods under different user-sensor distances. The Gaze error is expressed as the mean across the two directions θ and γ.

Tab.5.1 performs a global comparison of our gaze estimation systems to the state-of-the-art methods. Some methods report the results only under specific distances and frontal configurations.

5.6 Conclusion

In this chapter, we described the fully appearance-based approach for a robust and unconstrained gaze estimation. This approach is a promising solution to overcome the limitations of the existing systems described previously.

In the previous chapters, we highlighted the importance of the head pose component in automatic gaze estimation systems. Designing an optimal system which exploits both eyes and face cues was a challenging task. We proposed to use a multi-channel containing both information as training sample to robustly learn the mapping between gaze and appearance spaces using regression tree predictors.

To boost the ability of generalization of the trees, we used a 3D face morphable model with an integrated parametric gaze model to render RGB-D gaze samples. We described the integration of the gaze model to the 3DMM allowing us to generate sufficiently accurate gaze ground truth. In addition to synthetic RGB-D samples, we recorded real samples using Kinect sensor and described the followed protocol.

Different experiments were performed to evaluate the accuracy of this approach. A principal forest was trained using exclusively synthetic data and tested on real scenarios. The obtained results on unconstrained configurations demonstrate the great potential of this approach in handling gaze estimation. To illustrate the importance of each cue in the global system, we trained different forests on data targeting at each time a specific channel. The different obtained estimations showed the involvement of both RGB and depth information of the final estimation, especially for depth cue under strong head pose configuration. To evaluate the realism of our rendered training samples, we compare the estimation accuracy with a forest trained on real data and forests trained on different numbers of synthetic samples. We deducted an empirical ratio between the cardinal of real and synthetic set yielding comparable performance.

Finally, we compared this approach to the semi appearance-based approach which assumes and considers gaze information and head pose as two independent blocks. We describe the different forests used to perform an objective comparison. Testing gaze estimation under strong head pose changes showed that our fully appearance-based approach provides better results and meets the requirement of our environment in terms of accuracy and robustness. A global comparison of our systems and the state-of-art method was provided.

Chapter 6

Conclusions

Contents

6.1 Conclusions

In this work, we tackled automatic gaze estimation problem. Our context was about pointing objects displayed in real/virtual environment. Strong assumptions were established beforehand. The user is allowed to perform free movement including important head pose variation and large distances to the sensors. According to these unfavorable conditions, this work belongs on highly unconstrained gaze estimation systems.

During our investigation, we explored the existing approaches in gaze tracking and analysis field with a view to enhancing accuracy and performance. For each approach, we developed a specific automatic gaze estimation system and performed evaluation experiments. To overcome the limitations presented by these approaches, we established a novel paradigm and proposed a final automatic gaze estimation system. All the presented systems share a common dominator in their processing which is the involvement of Random Forest algorithm. These systems can be summarized as follows:

- **Feature-based system:** in this approach, we presented a system principally based on eye-pupil localization and head pose parameters estimation. To determine the gaze information, our method follows, as usually performed, a 3D eye model to calibrate accurately the user's eye features (eye centers, cornea center..etc.). To infer the 3D gaze information, we used the head pose component to project eye feature points into world coordinates system. To provide the potential of this approach, we compared separately each component to the state-of-art methods yielding a final evaluation.

- **Semi appearance-based system:** instead of considering the eye as a geometric model, this approach aims to infer the gaze information using the eye image as appearance. The problem is formulated as a regression one. Recent methods try to learn a robust mapping function between high dimensional eye image space and gaze space. One of the particular assumption of these methods is to consider only eye image under frontal configuration and to use head pose parameters as a geometric correctional tool to establish a final gaze estimation. We presented our system following this assumption and performed a comprehensive comparison to some recent methods.

- **Fully appearance-based system:** according to the results of the previous systems, we developed our final system which follows a novel paradigm. Admittedly, classical approaches tackle automatic gaze estimation problem by assuming eye and head pose parameters as two independent blocks. One direct consequence of such design is to cumulate error of each component in the final estimation. In our system, we proposed a way to unify these two blocs through a global gaze manifold with no geometric assumptions. We formulate the problem as a regression one by using Random Forest as a tool to learn the mapping. By using RGB conjointly with depth cues which encode eye appearance and head pose information respectively as input, we achieved an unified block producing a robust and efficient gaze estimation. This system meets our requirement related to the user environment in terms of distance to sensor, head pose changes, illumination variation and appearance variability.

Another highlight of this work was the importance of the data. Indeed, as we used Random Forest as a main tool to tackle learning problems through different systems especially for the last one, building a significant training dataset is fundamental to ensure robustness and generalization. Two global approaches related to the type of data exploited can be reported as follows:

- **Synthetic data approach:** the main motivation was about providing a very important amount of training samples automatically labeled in a reasonable time. We used a human face 3DMM able to synthesize different shapes and textures as an inter-user appearance variability. To provide such model with a supplementary ability of synthesizing different gaze directions, we integrated a dynamic gaze system. The final rendered gaze RGB-D samples demonstrated a great potential in handling of unconstrained gaze estimation task.

- **Real data approach:** by using the Kinect as depth sensor, we recorded a significant real RGB-D gaze database. We followed a rigorous protocol to record different participants and obtained reliable labeled samples after a tedious preprocessing step. This database allowed us to perform a comparison with synthetic data in terms of robustness and accuracy. We finished our experiments by establishing an empirical real/synthetic ratio producing similar performance.

6.2 Limitations and perspectives

During our investigation, some points have not been deepened sufficiently. Sometimes, they are related to the learning model, sometimes to the data. We consider that treating these points would enhance efficiency, we report them as follows:

- **Random Forest:** in this work, for each learning task, we kept the input data unchanged. For instance, in eye pupil localization, trees are learned on the raw gray intensities of the patches, in head pose estimation, trees learn from difference on integral raw depth intensities, in fully appearance, trees learn from both raw RGB and depth cues. Instead of keeping the same representation of the data, introducing a deep strategy to learn the optimal input features can enhance discriminative ability of the trees. Usually, this strategy is applied within neuronal network. [Kontschieder 2015] demonstrated an elegant way to conjointly learn the mapping and the data representation.

  We already demonstrated in our experiments the importance of depth information in the final estimation. To quantify the degree of involvement of each cue, a counting parameter in addition to the selection strategy at each node, would establish a histogram related to cues weighting at the leaves.

- **Synthetic data:** in this work we used a 3DMM exclusively designed for face reconstruction and recognition tasks. To adapt such model to gaze estimation, we introduced a simplified dynamic gaze model. For instance, we modeled the eyelids movements as uniform blendshapes. A more rigorous way would be to analyze eyelids points movement as a function of the gaze direction across significant participants. Such analysis allows more accurate modeling of the eyelids behavior and covers specific gazing scenarios.

  A radical solution would be to build a specific gaze 3DMM. With highly accurate scanning technique, different persons gazing targets in 2D/3D can be recorded, then applying a PCA for both shape/texture information would provide specific modes of variation related to gaze. Such modes would synthesize more naturally eyelids movements.

- **Real data:** in our work we obtained sufficient amount of real training samples to perform the comparison to synthetic data using Kinect v.1. The second version of this sensor presents considerable advantages in terms of data resolution and acquisition time. Using this version would allow larger user-sensor distance and allow to overcome the limitation related to computational time.

This work gives rise to another collaboration between Interactions Immersives and FAST teams. The main topic of this future work will be about 3D object estimation and camera re-localization. This new thesis would exploit the results of our work and, through our perspectives, would produce more successful results.

List of Figures

List of Tables

Appendix A

Appendix

A.1 Information gain for continuous distribution

The objective function (information gain) for each node j in the random forest framework is expressed as

$$E_j = H(\mathcal{P}_j) - \sum_{i \in \{\mathcal{L},\mathcal{R}\}} \frac{|\mathcal{P}_j^i|}{|\mathcal{P}_j|} H(\mathcal{P}_j^i)$$

The entropy H for a continuous variable y is defined as :

$$H(\mathcal{P}) = -\frac{1}{|\mathcal{P}|} \sum_{x \in \mathcal{P}} \int_y p(y|x) \log(p(y|x)) dy$$

By modeling $p(y|x)$ as a Gaussian distribution:

$$p(y|x) = \mathcal{N}(y, \bar{y}, \sigma^2)$$

An explicit description of $p(y|x)$ as a normal distribution $f(y)$ is expressed:

$$f(y) = \frac{1}{\sqrt[2]{2\pi\sigma^2}} \exp\{-\frac{(y-\bar{y})^2}{2\sigma^2}\}$$

we can rewrite the entropy as follows:

$$\begin{aligned}
H(\mathcal{P}) &= -\frac{1}{|\mathcal{P}|}\Big(\sum \int f(y) \log(\frac{1}{\sqrt[2]{2\pi\sigma^2}} \exp\{-\frac{(y-\bar{y})^2}{2\sigma^2}\})\Big) \\
&= -\frac{1}{|\mathcal{P}|}\Big(\sum_{\mathcal{P}} \int_y f(y)\big[\log(\exp\{-\frac{(y-\bar{y})^2}{2\sigma^2}\}) + \log(\frac{1}{\sqrt[2]{2\pi\sigma^2}})\big] dy\Big) \\
&= -\frac{1}{|\mathcal{P}|}\Big(\sum_{\mathcal{P}} \int_y f(y)\big[-\frac{(y-\bar{y})^2}{2\sigma^2} + \frac{1}{2}\log(2\pi\sigma^2)\big] dy\Big) \\
&= -\frac{1}{|\mathcal{P}|}\Big(\sum_{\mathcal{P}} \Big(\frac{1}{2\sigma^2} \int_y f(y)(y-\bar{y})^2 dy + \frac{1}{2}\log(2\pi\sigma^2) \int_y f(y) dy\Big)\Big) \\
&= -\frac{1}{|\mathcal{P}|}\Big(\sum_{\mathcal{P}} \Big(\frac{1}{2\sigma^2}\sigma^2 + \frac{1}{2}\log(2\pi\sigma^2)\Big)\Big) \\
&= -\frac{1}{|\mathcal{P}|}\Big(\sum_{\mathcal{P}} \Big(\frac{1}{2}\log(2\pi e\sigma^2)\Big)\Big)
\end{aligned}$$

Finally, the information gain can be expressed as follows:

$$E_j \propto \sum_{x \in \mathcal{P}} \log\left(\sigma_y(x)\right) - \sum_{i \in \{\mathcal{L},\mathcal{R}\}} \Big(\sum_{x \in \mathcal{P}_i} \log\left(\sigma_y(x)\right) \Big)$$

For a multivariate Gaussian distribution, we obtain the following formula:

$$E_j = \sum_{x \in \mathcal{P}} \log\left(|\Lambda_y(x)|\right) - \sum_{i \in \{\mathcal{L},\mathcal{R}\}} \Big(\sum_{x \in \mathcal{P}_i} \log\left(|\Lambda_y(x)|\right) \Big)$$

where $|\Lambda|$ represents the determinant of the covariance matrix.

Bibliography

[Ahn 2014] Byungtae Ahn, Jaesik Park and In So Kweon. *Real-time head orientation from a monocular camera using deep neural network.* In Asian Conference on Computer Vision, pages 82–96. Springer, 2014. (Cited on pages 5, 7 and 8.)

[Amberg 2008] Brian Amberg, Reinhard Knothe and Thomas Vetter. *Expression invariant 3D face recognition with a morphable model.* In Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on, pages 1–6. IEEE, 2008. (Cited on page 7.)

[Ashraf 2008] Ahmed Bilal Ashraf, Simon Lucey and Tsuhan Chen. *Learning patch correspondences for improved viewpoint invariant face recognition.* In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008. (Cited on page 55.)

[Asthana 2009] Akshay Asthana, Tom Gedeon, Roland Goecke and Conrad Sanderson. *Learning-based face synthesis for pose-robust recognition from single image.* In British Machine Vision Conference 2009, pages 1–10. British Machine Vision Association and Society for Pattern Recognition, 2009. (Cited on page 55.)

[Baltrušaitis 2014] Tadas Baltrušaitis, Peter Robinson and Louis-Philippe Morency. *Continuous conditional neural fields for structured regression.* In European Conference on Computer Vision, pages 593–608. Springer, 2014. (Cited on page 18.)

[Baltruvsaitis 2012] Tadas Baltruvsaitis, Peter Robinson and Louis-Philippe Morency. *3D constrained local model for rigid and non-rigid facial tracking.* In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2610–2617. IEEE, 2012. (Cited on page 7.)

[Baluja 1994] Shumeet Baluja and Dean Pomerleau. *Non-intrusive gaze tracking using artificial neural networks.* Technical report, DTIC Document, 1994. (Cited on page 11.)

[Bär 2012] Tobias Bär, Jan Felix Reuter and J Marius Zöllner. *Driver head pose and gaze estimation based on multi-template icp 3-d point cloud alignment.*

In Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, pages 1797–1802. IEEE, 2012. (Cited on page 10.)

[Beymer 1994] David J Beymer. *Face recognition under varying pose.* In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on, pages 756–761. IEEE, 1994. (Cited on pages 4 and 7.)

[Blanz 2003] Volker Blanz and Thomas Vetter. *Face recognition based on fitting a 3D morphable model.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 9, pages 1063–1074, 2003. (Cited on page 55.)

[Breiman 2001] Leo Breiman. *Random forests.* Machine learning, vol. 45, no. 1, pages 5–32, 2001. (Cited on page 20.)

[Bruske 1998] J Bruske, E Abraham-Mumm, J Pauli and G Sommer. *Head-pose estimation from facial images with subspace neural networks.* In Proc. of Int. Neural Network and Brain Conference, pages 528–531, 1998. (Cited on page 5.)

[Bulo 2014] Samuel Bulo and Peter Kontschieder. *Neural decision forests for semantic image labelling.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 81–88, 2014. (Cited on pages 30 and 33.)

[Canton-Ferrer 2006] Cristian Canton-Ferrer, Josep Ramon Casas and Montse Pardàs. *Head pose detection based on fusion of multiple viewpoint information.* In Multimodal Technologies for Perception of Humans, pages 305–310. Springer, 2006. (Cited on page 7.)

[Cappelli 2000] Raffaele Cappelli, A Erol, D Maio and D Maltoni. *Synthetic fingerprint-image generation.* In Pattern Recognition, 2000. Proceedings. 15th International Conference on, volume 3, pages 471–474. IEEE, 2000. (Cited on page 66.)

[Chen 2003] Longbin Chen, Lei Zhang, Yuxiao Hu, Mingjing Li and Hongjiang Zhang. *Head Pose Estimation using Fisher Manifold Learning.* In AMFG, pages 203–207, 2003. (Cited on page 5.)

[Chen 2008] Jixu Chen and Qiang Ji. *3D gaze estimation with a single camera without IR illumination.* In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pages 1–4. IEEE, 2008. (Cited on pages 10 and 84.)

[Cootes 2001] Timothy F Cootes, Gareth J Edwards and Christopher J Taylor. *Active appearance models.* IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 6, pages 681–685, 2001. (Cited on pages 10 and 48.)

[Criminisi 2009] Antonio Criminisi, Jamie Shotton and Stefano Bucciarelli. *Decision forests with long-range spatial context for organ localization in CT volumes.* In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 69–80. Citeseer, 2009. (Cited on page 28.)

[Criminisi 2011] A Criminisi, J Shotton and E Konukoglu. *Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning.* Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114, vol. 5, no. 6, page 12, 2011. (Cited on pages 25 and 29.)

[Dantone 2012] Matthias Dantone, Juergen Gall, Gabriele Fanelli and Luc Van Gool. *Real-time facial feature detection using conditional regression forests.* In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2578–2585. IEEE, 2012. (Cited on pages 13, 17, 30, 32 and 33.)

[Ding 2015] Changxing Ding and Dacheng Tao. *A comprehensive survey on pose-invariant face recognition.* arXiv preprint arXiv:1502.04383, 2015. (Cited on page 55.)

[Ebisawa 1998] Yoshinobu Ebisawa. *Improved video-based eye-gaze detection method.* Instrumentation and Measurement, IEEE Transactions on, vol. 47, no. 4, pages 948–955, 1998. (Cited on page 8.)

[Fanelli 2011] Gabriele Fanelli, Juergen Gall and Luc Van Gool. *Real time head pose estimation with random regression forests.* In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 617–624. IEEE, 2011. (Cited on pages 5, 7, 8, 24, 49, 51 and 66.)

[Fanelli 2013] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati and Luc Van Gool. *Random Forests for Real Time 3D Face Analysis.* Int. J. Comput. Vision, vol. 101, no. 3, pages 437–458, February 2013. (Cited on page 49.)

[Gall 2013] Juergen Gall and Victor Lempitsky. *Class-specific hough forests for object detection.* In Decision forests for computer vision and medical image analysis, pages 143–157. Springer, 2013. (Cited on pages 25 and 31.)

[Gao 2009] Hua Gao, Hazım Kemal Ekenel and Rainer Stiefelhagen. *Pose normalization for local appearance-based face recognition.* In Advances in Biometrics, pages 32–41. Springer, 2009. (Cited on page 55.)

[Geurts 2006] Pierre Geurts, Damien Ernst and Louis Wehenkel. *Extremely randomized trees.* Machine learning, vol. 63, no. 1, pages 3–42, 2006. (Cited on page 29.)

[González-Jiménez 2007] Daniel González-Jiménez and José Luis Alba-Castro. *Toward pose-invariant 2-d face recognition through point distribution models and*

facial symmetry. Information Forensics and Security, IEEE Transactions on, vol. 2, no. 3, pages 413–429, 2007. (Cited on page 55.)

[Gray 2013] KR Gray, P Aljabar, RA Heckemann, A Hammers and D Rueckert. *Manifold Forests for Multi-modality Classification of Alzheimer's Disease.* In Decision Forests for Computer Vision and Medical Image Analysis, pages 261–272. Springer, 2013. (Cited on page 25.)

[Guestrin 2006] Elias Daniel Guestrin and Moshe Eizenman. *General theory of remote gaze estimation using the pupil center and corneal reflections.* Biomedical Engineering, IEEE Transactions on, vol. 53, no. 6, pages 1124–1133, 2006. (Cited on pages 4, 11, 40 and 51.)

[Hansen 2002] Dan Witzner Hansen, John Paulin Hansen, Mads Nielsen, Anders Sewerin Johansen and Mikkel B Stegmann. *Eye typing using Markov and active appearance models.* In Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on, pages 132–136. IEEE, 2002. (Cited on page 11.)

[Hansen 2007] Dan Witzner Hansen and Riad I Hammoud. *An improved likelihood model for eye tracking.* Computer Vision and Image Understanding, vol. 106, no. 2, pages 220–230, 2007. (Cited on page 8.)

[Hansen 2010] Dan Witzner Hansen and Qiang Ji. *In the eye of the beholder: A survey of models for eyes and gaze.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 3, pages 478–500, 2010. (Cited on page 2.)

[Haro 2000] Antonio Haro, Myron Flickner and Irfan Essa. *Detecting and tracking eyes by using their physiological properties, dynamics, and appearance.* In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, volume 1, pages 163–168. IEEE, 2000. (Cited on page 8.)

[Heinzmann 1998] Jochen Heinzmann and Alexander Zelinsky. *3-D facial pose and gaze point estimation using a robust real-time tracking paradigm.* In Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, pages 142–147. IEEE, 1998. (Cited on page 7.)

[Ho 2013] Huy Tho Ho and Rama Chellappa. *Pose-invariant face recognition using markov random fields.* Image Processing, IEEE Transactions on, vol. 22, no. 4, pages 1573–1584, 2013. (Cited on page 55.)

[Horprasert 1997] Thanarat Horprasert, Yaser Yacoob and Larry S Davis. *Computing 3D head orientation from a monocular image sequence.* In 25th Annual AIPR Workshop on Emerging Applications of Computer Vision, pages 244–252. International Society for Optics and Photonics, 1997. (Cited on page 7.)

[Hu 2004] Changbo Hu, Jing Xiao, Iain Matthews, Simon Baker, Jeffrey F Cohn and Takeo Kanade. *Fitting a Single Active Appearance Model Simultaneously to Multiple Images.* In BMVC, pages 1–10, 2004. (Cited on page 6.)

[Huang 1998] Jeffrey Huang, Xuhui Shao and Harry Wechsler. *Face pose discrimination using support vector machines (SVM).* In Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on, volume 1, pages 154–156. IEEE, 1998. (Cited on page 5.)

[Huang 2004] Kohsia S Huang and Mohan M Trivedi. *Robust real-time detection, tracking, and pose estimation of faces in video streams.* In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 3, pages 965–968. IEEE, 2004. (Cited on page 7.)

[Huang 2007] Gary B Huang, Manu Ramesh, Tamara Berg and Erik Learned-Miller. *Labeled faces in the wild: A database for studying face recognition in unconstrained environments.* Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. (Cited on page 13.)

[Ishikawa 2004] Takahiro Ishikawa. *Passive driver gaze tracking with active appearance models.* 2004. (Cited on page 10.)

[Jang 2008] Jun-Su Jang and Takeo Kanade. *Robust 3d head tracking by online feature registration.* In 8th IEEE Int. Conf. on Automatic Face and Gesture Recognition. Citeseer, 2008. (Cited on page 7.)

[Jesorsky 2001] Oliver Jesorsky, Klaus J Kirchberg and Robert W Frischholz. *Robust face detection using the hausdorff distance.* In Audio-and video-based biometric person authentication, pages 90–95. Springer, 2001. (Cited on pages 45 and 46.)

[Ji 2002] Qiang Ji and Xiaojie Yang. *Real-time eye, gaze, and face pose tracking for monitoring driver vigilance.* Real-Time Imaging, vol. 8, no. 5, pages 357–377, 2002. (Cited on page 8.)

[Jianfeng 2014] Li Jianfeng and Li Shigang. *Eye-model-based gaze estimation by rgb-d camera.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 592–596, 2014. (Cited on pages 10, 52 and 84.)

[Jones 2003] Michael Jones and Paul Viola. *Fast multi-view face detection.* Mitsubishi Electric Research Lab TR-20003-96, vol. 3, page 14, 2003. (Cited on page 5.)

[Kacete 2016] Amine Kacete, Renaud Seguier, JerÃ´me Royan, Michel Collobert and Catherine Soladie. *Real-time eye pupil localization using Hough regression forest.* In Applications of Computer Vision, 2016.(WACV 2016). Proceedings. Sixth IEEE Workshop on. IEEE, 2016. (Cited on page 30.)

[Kohlbecher 2008] Stefan Kohlbecher, Stanislavs Bardinst, Klaus Bartl, Erich Schneider, Tony Poitschke and Markus Ablassmeier. *Calibration-free eye tracking by reconstruction of the pupil ellipse in 3D space*. In Proceedings of the 2008 symposium on Eye tracking research & applications, pages 135–138. ACM, 2008. (Cited on page 10.)

[Kontschieder 2015] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi and Samuel Rota Bulo. *Deep Neural Decision Forests*. In Proceedings of the IEEE International Conference on Computer Vision, pages 1467–1475, 2015. (Cited on pages 30, 34 and 88.)

[Krüger 1997] Norbert Krüger, Michael Pötzsch and Christoph von der Malsburg. *Determination of face position and pose with a learned representation based on labelled graphs*. Image and vision computing, vol. 15, no. 8, pages 665–673, 1997. (Cited on pages 6 and 7.)

[La Cascia 2000] Marco La Cascia, Stan Sclaroff and Vassilis Athitsos. *Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 4, pages 322–336, 2000. (Cited on page 7.)

[Lefèvre 2009] Stéphanie Lefèvre and Jean-Marc Odobez. *Structure and appearance features for robust 3d facial actions tracking*. In Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on, pages 298–301. IEEE, 2009. (Cited on page 7.)

[Lepetit 2005] Vincent Lepetit, Pascal Lagger and Pascal Fua. *Randomized trees for real-time keypoint recognition*. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 2, pages 775–781. IEEE, 2005. (Cited on page 23.)

[Li 2000] Yongmin Li, Shaogang Gong and Heather Liddell. *Support vector regression and classification based multi-view face detection and recognition*. In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pages 300–305. IEEE, 2000. (Cited on page 5.)

[Li 2012] Shaoxin Li, Xin Liu, Xiujuan Chai, Haihong Zhang, Shihong Lao and Shiguang Shan. *Morphable displacement field based image matching for face recognition across pose*. In Computer Vision–ECCV 2012, pages 102–115. Springer, 2012. (Cited on page 55.)

[Lowe 1999] David G Lowe. *Object recognition from local scale-invariant features*. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. Ieee, 1999. (Cited on page 9.)

[Lu 2011a] Feng Lu, Takahiro Okabe, Yusuke Sugano and Yoichi Sato. *A Head Pose-free Approach for Appearance-based Gaze Estimation*. In BMVC, pages 1–11, 2011. (Cited on pages 11, 12, 53 and 54.)

[Lu 2011b] Feng Lu, Yusuke Sugano, Takahiro Okabe and Yoichi Sato. *Inferring human gaze from appearance via adaptive linear regression.* In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 153–160. IEEE, 2011. (Cited on pages 11 and 54.)

[Marée 2007] Raphaël Marée, Pierre Geurts and Louis Wehenkel. *Random subwindows and extremely randomized trees for image classification in cell biology.* BMC Cell Biology, vol. 8, no. Suppl 1, page S2, 2007. (Cited on pages 23 and 30.)

[Markuš 2014] Nenad Markuš, Miroslav Frljak, Igor S Pandžić, Jörgen Ahlberg and Robert Forchheimer. *Eye pupil localization with an ensemble of randomized trees.* Pattern recognition, vol. 47, no. 2, pages 578–587, 2014. (Cited on pages 9, 10, 42, 43, 46, 47 and 48.)

[Matsumoto 2000] Yoshio Matsumoto and Alexander Zelinsky. *An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement.* In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pages 499–504. IEEE, 2000. (Cited on page 10.)

[McKenna 1998] Stephen J McKenna and Shaogang Gong. *Real-time face pose estimation.* Real-Time Imaging, vol. 4, no. 5, pages 333–347, 1998. (Cited on page 5.)

[Milborrow 2010] S. Milborrow, J. Morkel and F. Nicolls. *The MUCT Landmarked Face Database.* Pattern Recognition Association of South Africa, 2010. (Cited on pages 13 and 15.)

[Moosmann 2007] Frank Moosmann, Bill Triggs and Frederic Jurie. *Fast discriminative visual codebooks using randomized clustering forests.* In Twentieth Annual Conference on Neural Information Processing Systems (NIPS'06), pages 985–992. MIT Press, 2007. (Cited on page 25.)

[Mora 2012] Kenneth Alberto Funes Mora and Jean-Marc Odobez. *Gaze estimation from multimodal kinect data.* In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pages 25–30. IEEE, 2012. (Cited on pages 7, 11, 12, 52, 53, 54, 58, 59 and 84.)

[Mora 2014] Kenneth Alberto Funes Mora, Florent Monay and Jean-Marc Odobez. *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, pages 255–258. ACM, 2014. (Cited on page 15.)

[Morency 2002] Louis-Philippe Morency, Ali Rahimi, Neal Checka and Trevor Darrell. *Fast stereo-based head tracking for interactive environments.* In Auto-

matic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, pages 390–395. IEEE, 2002. (Cited on page 7.)

[Morency 2003] Louis-Philippe Morency, Ali Rahimi and Trevor Darrell. *Adaptive view-based appearance models.* In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 1, pages I–803. IEEE, 2003. (Cited on page 7.)

[Morency 2010] Louis-Philippe Morency, Jacob Whitehill and Javier Movellan. *Monocular head pose estimation using generalized adaptive view-based appearance model.* Image and Vision Computing, vol. 28, no. 5, pages 754–761, 2010. (Cited on page 7.)

[Morimoto 2000] Carlos H Morimoto and Myron Flickner. *Real-time multiple face detection using active illumination.* In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pages 8–13. IEEE, 2000. (Cited on page 8.)

[Morimoto 2005] Carlos H Morimoto and Marcio RM Mimica. *Eye gaze tracking techniques for interactive applications.* Computer Vision and Image Understanding, vol. 98, no. 1, pages 4–24, 2005. (Cited on page 11.)

[Murphy-Chutorian 2009] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. *Head pose estimation in computer vision: A survey.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 4, pages 607–626, 2009. (Cited on pages 4 and 7.)

[Newman 2000] Rhys Newman, Yoshio Matsumoto, Sebastien Rougeaux and Alexander Zelinsky. *Real-time stereo tracking for head pose and gaze estimation.* In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pages 122–128. IEEE, 2000. (Cited on page 7.)

[Niyogi 1996] Sourabh Niyogi and William T Freeman. *Example-based head tracking.* In Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on, pages 374–378. IEEE, 1996. (Cited on page 4.)

[Oka 2005] Kenji Oka, Yoichi Sato, Yasuto Nakanishi and Hideki Koike. *Head Pose Estimation System Based on Particle Filtering with Adaptive Diffusion Control.* In MVA, pages 586–589, 2005. (Cited on page 7.)

[Olshen 1984] LBJFR Olshen, Charles J Stone*et al.* *Classification and regression trees.* Wadsworth International Group, vol. 93, no. 99, page 101, 1984. (Cited on page 20.)

[Ozuysal 2010] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit and Pascal Fua. *Fast keypoint recognition using random ferns.* IEEE transactions on

pattern analysis and machine intelligence, vol. 32, no. 3, pages 448–461, 2010. (Cited on page 30.)

[Padeleris 2012] Pashalis Padeleris, Xenophon Zabulis and Antonis A Argyros. *Head pose estimation on depth data based on Particle Swarm Optimization.* In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pages 42–49. IEEE, 2012. (Cited on pages 5 and 7.)

[Papazov 2015] Chavdar Papazov, Tim K Marks and Michael Jones. *Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4722–4730, 2015. (Cited on pages 5, 7 and 8.)

[Pauly 2011] Olivier Pauly, Ben Glocker, Antonio Criminisi, Diana Mateus, Axel Martinez Möller, Stephan Nekolla and Nassir Navab. *Fast multiple organ detection and localization in whole-body MR Dixon sequences.* In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 239–247. Springer, 2011. (Cited on page 31.)

[Paysan 2009] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani and Thomas Vetter. *A 3D face model for pose and illumination invariant face recognition.* In Advanced Video and Signal Based Surveillance, 2009. (Cited on pages 65, 66, 67 and 68.)

[Quinlan 1993] J Ross Quinlan. C4. 5: programs for machine learning. 1993. (Cited on page 20.)

[Ram 2011] Parikshit Ram and Alexander G Gray. *Density estimation trees.* In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 627–635. ACM, 2011. (Cited on page 25.)

[Rowley 1998] Henry A Rowley, Shumeet Baluja and Takeo Kanade. *Rotation invariant neural network-based face detection.* In Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pages 38–44. IEEE, 1998. (Cited on page 5.)

[Schapire 1990] Robert E Schapire. *The strength of weak learnability.* Machine learning, vol. 5, no. 2, pages 197–227, 1990. (Cited on page 20.)

[Sherrah 1999] Jamie Sherrah, Shaogang Gong and Eng-Jon Ong. *Understanding Pose Discrimination in Similarity Space.* In BMVC, pages 1–10, 1999. (Cited on page 5.)

[Shotton 2013] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook and Richard Moore. *Real-time human pose recognition in parts from single depth images.* Communications

of the ACM, vol. 56, no. 1, pages 116–124, 2013. (Cited on pages 20, 24, 25 and 66.)

[Smith 2013] Brian A Smith, Qi Yin, Steven K Feiner and Shree K Nayar. *Gaze locking: passive eye contact detection for human-object interaction.* In Proceedings of the 26th annual ACM symposium on User interface software and technology, pages 271–280. ACM, 2013. (Cited on page 14.)

[Stiefelhagen 2004] Rainer Stiefelhagen. *Estimating head pose with neural networks-results on the pointing04 icpr workshop evaluation data.* In Pointing'04 ICPR Workshop of the Int. Conf. on Pattern Recognition, 2004. (Cited on pages 5 and 7.)

[Sugano 2008] Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato and Hideki Koike. *An incremental learning method for unconstrained gaze estimation.* In Computer Vision–ECCV 2008, pages 656–667. Springer, 2008. (Cited on page 11.)

[Sugano 2010] Yusuke Sugano, Yasuyuki Matsushita and Yoichi Sato. *Calibration-free gaze sensing using saliency maps.* In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 2667–2674. IEEE, 2010. (Cited on page 11.)

[Sun 2012] Min Sun, Pushmeet Kohli and Jamie Shotton. *Conditional regression forests for human pose estimation.* In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 3394–3401. IEEE, 2012. (Cited on pages 32 and 33.)

[Tan 2002] Kar-Han Tan, David J Kriegman and Narendra Ahuja. *Appearance-based eye gaze estimation.* In Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on, pages 191–195. IEEE, 2002. (Cited on page 11.)

[Thian 2003] Norman Poh Hoon Thian, Sébastien Marcel and Samy Bengio. *Improving face authentication using virtual samples.* In Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, volume 3, pages III–233. IEEE, 2003. (Cited on page 66.)

[Timm 2011] Fabian Timm and Erhardt Barth. *Accurate Eye Centre Localisation by Means of Gradients.* VISAPP, vol. 11, pages 125–130, 2011. (Cited on pages 9, 10 and 46.)

[Valenti 2008] Roberto Valenti and Theo Gevers. *Accurate eye center location and tracking using isophote curvature.* In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008. (Cited on pages 9, 10 and 46.)

[Valenti 2012] Roberto Valenti, Nicu Sebe and Theo Gevers. *Combining head pose and eye location information for gaze estimation.* Image Processing, IEEE Transactions on, vol. 21, no. 2, pages 802–815, 2012. (Cited on page 10.)

[Vetter 1998] Thomas Vetter. *Synthesis of novel views from a single face image.* International journal of computer vision, vol. 28, no. 2, pages 103–116, 1998. (Cited on page 55.)

[Villanueva 2013] Arantxa Villanueva, Victoria Ponz, Laura Sesma-Sanchez, Mikel Ariz, Sonia Porta and Rafael Cabeza. *Hybrid method based on topography for robust detection of iris center and eye corners.* ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 9, no. 4, page 25, 2013. (Cited on pages 12 and 42.)

[Viola 2001] Paul Viola and Michael Jones. *Rapid object detection using a boosted cascade of simple features.* In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–511. IEEE, 2001. (Cited on pages 9 and 20.)

[Viola 2004] Paul Viola and Michael J Jones. *Robust real-time face detection.* International journal of computer vision, vol. 57, no. 2, pages 137–154, 2004. (Cited on page 57.)

[Voit 2005] Michael Voit, Kai Nickel and Rainer Stiefelhagen. *Neural Network-based Head Pose Estimation and Multi-view Fusion–Draft Version–*, 2005. (Cited on pages 5 and 7.)

[Wang 2002] Jian-Gang Wang and Eric Sung. *Study on eye gaze estimation.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 32, no. 3, pages 332–350, 2002. (Cited on page 10.)

[Wang 2012] Haibo Wang, Franck Davoine, Vincent Lepetit, Christophe Chaillou and Chunhong Pan. *3-d head tracking via invariant keypoint learning.* Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 8, pages 1113–1126, 2012. (Cited on page 7.)

[Weidenbacher 2007] U Weidenbacher, G Layher, P-M Strauss and H Neumann. *A comprehensive head pose and gaze database.* In Intelligent Environments, 2007. IE 07. 3rd IET International Conference on, pages 455–458. IET, 2007. (Cited on pages 14 and 42.)

[Weise 2011] Thibaut Weise, Sofien Bouaziz, Hao Li and Mark Pauly. *Realtime performance-based facial animation.* In ACM Transactions on Graphics (TOG), volume 30, page 77. ACM, 2011. (Cited on page 7.)

[Williams 2006] Oliver Williams, Andrew Blake and Roberto Cipolla. *Sparse and Semi-supervised Visual Mapping with the S^ 3GP.* In Computer Vision and

Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 1, pages 230–237. IEEE, 2006. (Cited on page 11.)

[Wilson 2000] Hugh R Wilson, Frances Wilkinson, Li-Ming Lin and Maja Castillo. *Perception of head orientation.* Vision research, vol. 40, no. 5, pages 459–472, 2000. (Cited on page 6.)

[Wu 2008] Junwen Wu and Mohan M Trivedi. *A two-stage head pose estimation framework and evaluation.* Pattern Recognition, vol. 41, no. 3, pages 1138–1158, 2008. (Cited on pages 5 and 7.)

[Xiong 2005] Yingen Xiong and Francis Quek. *Meeting room configuration and multiple camera calibration in meeting analysis.* In Proceedings of the 7th international conference on Multimodal interfaces, pages 37–44. ACM, 2005. (Cited on page 7.)

[Xiong 2013] Xuehan Xiong and Fernando Torre. *Supervised descent method and its applications to face alignment.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 532–539, 2013. (Cited on page 17.)

[Xu 1998] Li-Qun Xu, Dave Machin and Phil Sheppard. *A Novel Approach to Real-time Non-intrusive Gaze Finding.* In BMVC, pages 1–10, 1998. (Cited on page 11.)

[Yang 2002] Ruigang Yang and Zhengyou Zhang. *Model-based head pose tracking with stereovision.* In Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, pages 255–260. IEEE, 2002. (Cited on page 7.)

[Zhang 2015] Xucong Zhang, Yusuke Sugano, Mario Fritz and Andreas Bulling. *Appearance-based gaze estimation in the wild.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4511–4520, 2015. (Cited on pages 11, 12 and 15.)

[Zhao 2002] Liang Zhao, Gopal Pingali and Ingrid Carlbom. *Real-time head orientation estimation using neural networks.* In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–297. IEEE, 2002. (Cited on page 5.)

[Zhu 2002] Zhiwei Zhu, Qiang Ji, Kikuo Fujimura and Kuangchih Lee. *Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination.* In Pattern Recognition, 2002. Proceedings. 16th International Conference on, volume 4, pages 318–321. IEEE, 2002. (Cited on page 8.)

[Zhu 2007] Zhiwei Zhu and Qiang Ji. *Novel eye gaze tracking techniques under natural head movement.* Biomedical Engineering, IEEE Transactions on, vol. 54, no. 12, pages 2246–2260, 2007. (Cited on page 11.)

[Zuo 2007] Jinyu Zuo, Natalia A Schmid and Xiaohan Chen. *On generation and analysis of synthetic iris images.* Information Forensics and Security, IEEE Transactions on, vol. 2, no. 1, pages 77–90, 2007. (Cited on page 66.)