Joint Head Tracking and Pose Estimation for Visual Focus of Attention Recognition

THÈSE N^o 3764 (2007)

présentée à la Faculté des Sciences et Techniques de l'Ingénieur

Laboratoire de l'IDIAP

Section de Génie Electrique et Electronique

École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de docteur ès sciences

par

Silèye Oumar Ba

Diplome d'Etudes Approfondies de Mathématiques, Vision et Apprentissage,
Ecole Normale Supérieure de Cachan, Paris, France
et de nationalité sénégalaise

acceptée sur proposition du jury:

Prof. Josip Mosig, EPFL, Président du jury

Prof. Hervé Bourlard, **Dr. Jean-Marc Odobez**, IDIAP Research Institute, directeurs de thèse

Prof. Maja Pantic, Twente University, The Netherland, rapporteur

Dr. Rainer Stiefelhagen, Karlsruhe University, Germany, rapporteur

Prof. Jean-Philippe Thiran, EPFL, Switzerland, rapporteur

Ecole Polytechnique Fédérale (EPFL) de Lausanne, Switzerland

28 Février 2007

Abstract

During the last two decades, computer science what are the ability to give to provide to machines in order to give them the ability to understand human behavior. One of them which is an important key to understand human behaviors, is the visual focus of attention (VFOA) of a person, which can be inferred from the gaze direction. The VFOA of a person gives insight about information such as who or what is the interest of a person, who is the target of the person's speech, who the person is listening to. Our interest in this thesis is to study people's VFOA using computer vision techniques. To estimate the VFOA of attention from a computer vision point of view, it is required to track the person's gaze. Because, tracking the eye gaze is impossible on low or mid resolution images, head orientation can be used as a surrogate for the gaze direction. Thus, in this thesis, we investigate in a first step the tracking of people's head orientation, in a second step the recognition of their VFOA from their head orientation.

For the head tracking, we consider probabilistic methods based on sequential Monte Carlo (SMC) techniques. The head pose space is discretized into a finite set of poses, and Multi-dimensional Gaussian appearance models are learned for each discrete pose. The discrete head models are embedded into a mixed state particle filter (MSPF) framework to jointly estimate the head location and pose. The evaluation shows that this approach works better than the traditional paradigm in which the head is first tracked then the head pose is estimated.

An important contribution of this thesis is the head pose tracking evaluation. As people usually evaluate their head pose tracking methods either qualitatively or with private data, we built a head pose video database using a magnetic field 3D location and orientation tracker. The database was used to evaluate our tracking methods, and was made publicly available to allow other researchers to evaluate and compare their algorithms.

Once the head pose is available, the recognition of the VFOA can be done. Two environments are considered to study the VFOA: a meeting room environment and an outdoor environment. In the meeting room environment, people are static. People's VFOAs were studied depending on their locations in the meeting room. The set of VFOAs for a person is discretized into a finite set of targets: the other people attending the meeting, the table, the slide screen, and another VFOA target called un-focused denoting that the person is focusing none of the previous defined VFOAs. The head poses are used as observations and potential VFOA targets as hidden states in a Gaussian mixture model (GMM) or a hidden Markov model (HMM) framework. The parameters of the emission probability distributions were learned by two ways. A first way using head pose training data, and a second way exploiting the geometry of the room and the head and eye-in-head rotations. Maximum a posteriori adaptation (MAP) of the VFOA models was to the input test data to take into account people personal ways of gazing at VFOA targets.

In the outdoor environment, people are moving and there is a single VFOA target. The problem in this study is to track multiple people passing and estimate whether or not they were focusing the advertisement. The VFOA is modeled as a GMM having as observations people's head location and pose.

Keywords: head pose tracking, sequential Monte Carlo, particle filters, Rao-Blackwellisation, Markov chain Monte Carlo, visual focus of attention estimation.

Résumé

Durant ces deux dernières décennies, les chercheurs ont travaillé sur des méthodes pour donner aux machines la capacité d'interagir avec les hommes. Un aspect important pour la compréhension du comportement humain par les machines est le centre de l'attention visuelle d'une personne. Le centre de l'attention visuelle peut être estimée à partir de la direction du regard. Le centre de l'attention visuelle d'une personne donne des informations quant à l'intérêt de la personne, quelle est la cible de son discours, qui est la personne qu'elle écoute. Dans cette thèse, notre but est l'étude du centre d'attention visuelle d'une personne à l'aide de méthodes basées sur la vision par ordinateur. Pour estimer le centre d'attention visuelle d'une personne, en vision par ordinateur, il est nécessaire de pouvoir suivre la direction du regard de la personne. Mais, quand sur des images de basses à moyennes résolutions, estimer la direction du regard d'une personne est impossible, l'oriention de la tête peut être utilisée comme palliatif à la direction du regard. Dès lors, nous nous devons de suivre l'orientation de la tête d'une personne, et, à partir de cette orientation, estimer le centre de son attention visuelle.

Dans cette thèse, nous étudions le suivi de l'oriention de la tête d'une personne à l'aide de modèles probabilistes basés sur des méthodes de Monté Carlo séquentielles. L'espace définissant l'oriention de la tête d'une personne est discrétisé en un ensemble fini. Des modèles d'apparences Gaussiennes sont appris pour chaque éliment de l'espace discret. Les modèles d'apparences sont utilisés, á travers une méthodologie de filtre particulaire à espace d'état mixte, pour une estimation jointe de la position et de l'orientation de la tête.

Un aspect important de cette thèse est l'évaluation des méthodes de suivis d'orientation de la tête. Parce que, en général, les chercheurs évaluent leur méthodes soit qualitativement, soit sur des bases de données d'évaluation privées, nous avons construit une base de données publique à l'aide d'un capteur électronique capable de suivre et de déterminer la location et l'oriention d'une tête. Nous utilisons alors cette base de donnée publique pour l'évaluation des nos méthodes de suivis.

Dès lors qu'est disponible l'orientation de la tête d'une personne, son centre d'attention visuelle peut être modélisé. L'orientation de la tête de la personne est alors utilisée comme observation, et les centres d'attention visuelles potentiels comme états cachés pour des modèles de mélanges de Gaussiennes ou des modèles à états cachés. Deux environnements seront considérés pour l'étude du centre d'intérêt visuel, une salle de réunion et une rue passante. Dans la salle de réunion, les personnes sont supposées assises à des places fixes. Pour une personne donnée, L'ensemble des centre d'attention visuelles possibles est défini comme étant: les autres personnes participant à la réunion, la table de la salle de réunion, l'écran de projection et un autre centre d'intérêt "non intéressé" signifiant que la personne ne s'intéresse à aucun des centre d'attentions visuelles pre-cités. pour la situation de la rue passante, les personnes sont potentiellement mobiles. Il existe alors un centre d'attention visuelle unique, une affiche sur une baie vitrée. Le but de notre étude est d'estimer si les passants portent ou non un intérêt visuel à l'affiche.

Mots Clés: algorithmes de suivi d'objets, méthodes de Monté Carlo séquentielles, filtres particulaires, Rao-Blackwellisation, chaînes de Markov, estimation du centre d'attention visuelle.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objectives and motivations

Nowadays, electronic devices are more and more present in human life. But, despite their ubiquitous presence in human life, electronic devices still need to be person-driven because they are scarcely equipped with human understanding capabilities. In the last years, vast investigations have been conducted by research fields such as artificial intelligence, speech processing or computer vision about the ways through which humans can interact with machines. Beyond classical ways human machine interaction based on device buttons or keyboards, other means of interactions have been investigated such as vocal interactions based on speech recognition and synthesis. In most cases, the interactions between humans and machines assumed people giving commands to a device or a computer. In other situations, the sensing devices have a passive role such as a microphone recording a speaker or a lecturer in a classroom, or a camera recording people and cars passing in a street. In these situations, people are not directly interacting with the sensing devices. Rather, from peoples' actions or interactions, the computer has to build an understanding of the occurring events. Thus, using a microphone, the goal can be to localize and recognize the speaker, to recognize the speech, to catch important keywords, to diarize the discourse. From video data, the goal can be to identify the people, track them through the physical space and recognize their actions. For instance, in a video surveillance task, actions to recognize can correspond to people abandoning luggage in public spaces such as airports or train stations.

In human actions understanding , human-human interaction plays an important role. Psychologists have been widely studying interactions in groups such as families or work teams. With the increasing availability of microphones and cameras, computer science researchers and psychologists are more and more investigating the analysis of human interaction in groups. Studying interactions between humans using a computer is by far more complex than studying human interactions with computers. In human-computer interaction, people behave in ways to be understood by the computer. For example, in a speech recognition situation, people will control their voice tone, and use specific keywords to make the computer understand the beginning and end of commands. In interactions between humans, people behavior is completely natural, and thus is less restricted because the recipient of the communication is another human.

Human interactions happen through verbal or non verbal cues. On one hand, the use of verbal cues is better understood because it is tightly connected to the taught explicit rules of language (grammar, dialog acts). On the other hand, the usage of non verbal cues is usually more implicit, which does not prevent it from following rules and exhibiting specific patterns in conversations. For instance, in a meeting context, a person rising a hand often means that he is requesting the floor, and a listener's head nod or shake can be interpreted as agreement or disagreement. Besides hand and head gestures, the VFOA is another important non verbal communication cue with functions such as establishing relationship (through mutual gaze), regulating the course of interaction, expressing intimacy, and exercising social control. A speaker's gaze often correlates with his addressees, i.e. the intended recipients of the speech, especially at a sentence end where the gaze can be interpreted as a request of back-channel. Also, for a listener, mutual gaze with the speaker are used to find appropriate time windows for speaker turn requests. Furthermore, studies have shown that a person's VFOA is influenced by the VFOA of other people [1]. Thus, recognizing the VFOA patterns of a group of people can reveal important knowledge about the participants' role and status, such as their influence, and the social nature of the occurring interactions.

Gaze is not interesting only from an "application" point of view, but also from a scientific point of view. Analyzing people's gaze requires the development of robust eye tracking systems as well as the modeling of the gaze itself. In computer vision, assuming the availability of high resolution images, tracking the eye gaze is feasible. However, the use of high resolution imagery is constraining when one needs to record people in large spaces. This constraints people either to remain close to the camera or to be static if they are far from the camera since in this case they need to be recorded with small camera field of views. In addition, from a psychologic point of view, gaze is strongly tied to the head orientation [1]. Human beings use three keys to estimate the gaze of another person: first the eye direction, secondly the head orientation, and thirdly the shoulder orientation. The eyes are used in priority, but the overall gaze direction is estimated based on the relationship between the three keys. Thus, when the eye gaze is not directly measurable information about the gaze can still be obtained from the head pose. This is what is done for gaze estimation when dealing with low resolution images in which the eyes are not visible enough.

1.2 Main contributions of the thesis

In this thesis, we investigated the recognition of the VFOA from head pose. To this end, we built a head pose database for head pose tracking evaluation, proposed probabilistic head pose tracking methods, and investigated VFOA recognition from head pose for static people in meeting situations, and for moving people in outdoor situations.

1.2.1 Head pose video database

Head pose tracking algorithms need to be evaluated. Although computer vision researchers have working on head pose tracking since at least a decade, there is a lack of public head pose evaluation database. Most head pose tracking systems are either evaluated qualitatively, or, in few cases numerically, using private databases. Thus, comparing head pose tracking algorithms is impossible. As a first contribution of this thesis, to fill this lack, we built a head pose tracking database using a magnetic field 3D location and orientation sensor tracker. The recordings were done in a meeting room environment and in an office environment. Figure 1.1 shows images from the database. This database has been made public to allow the researchers to evaluate the performances of their own head pose tracking systems. And, the database

(a) Meeting room data. (b) Office data.

Figure 1.1: Data from the head pose evaluation database. To people's, head the magnetic field location and orientation tracker used to annotate head poses is attached .

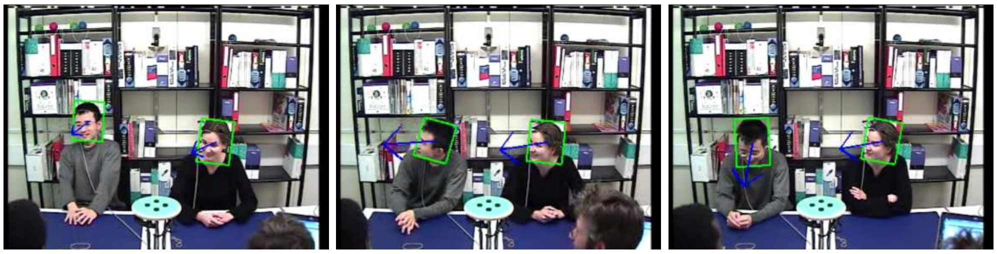
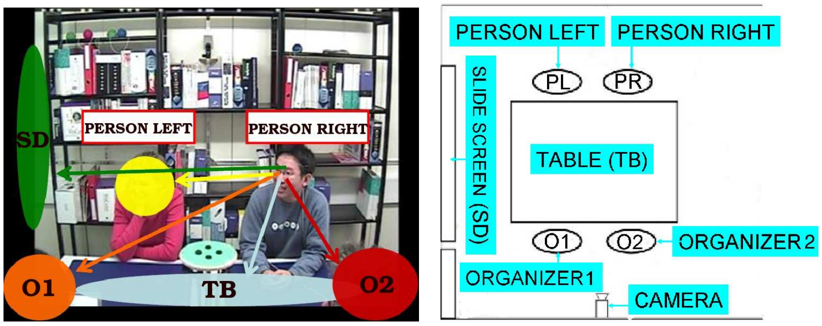


Figure 1.2: Head pose tracking results. The green boxes indicate estimated head location. The blue arrows indicate estimated head pose.

can be used to rank head pose tracking systems according to their performances. In addition, the head pose ground truth from the database can be used to have an insight of the optimal performances that can be expected when using head pose on other tasks.

1.2.2 Head pose tracking

To perform head pose tracking, we proposed a joint head tracking and pose estimation framework. We adopted a tracker that estimates jointly the head location and pose, contrarily to other researchers who preferred to track the head with a generic tracker then estimate its pose. In our methodology, the head pose space is first discretized and for each discrete pose value an appearance model is built using texture observations, skin color observations and binary observations from background subtraction. The the appearance models are then embedded into a mixed state particle filter allowing to represent jointly into a single state space model the head 2D image spatial configuration (location and size) and the 3D pose represented by a discrete variable labeling one of the element of the discretized space of head poses. Temporal head state estimation is done by recursive estimation of the posterior probability distribution (pdf) of the state given the observations through sequential Monte Carlo (SMC) techniques. The first SMC technique is an importance sampling particle filter (ISPF). The second technique is a Rao-Blackwellized version of the ISPF method. Rao-Blackwelization can be applied to a ISPF when the pdf of some components of the state variable can be computed exactly given the other variables. This is our case for the



(a) Potential VFOA targets. (b) Meeting room geometric configuration.

Figure 1.3: Recognizing people's VFOA. Figure 1.3(a) displays the potential VFOA targets for the right person, Figure 1.3(b) displays the geometric configuration of the room.

head pose variable because their set is discrete and finite. Rao-Blackwellization leads to better estimation performance than the standard ISPF method. Figure 1.2 shows samples images of head pose tracking results using the Rao-Blackwellized methodology. The third SMC method we investigated is based on Markov chain Monte Carlo (MCMC) sampling. The main contributions of our head pose tracking investigations are the following. First, we proposed a probabilistic framework for joint head tracking and pose estimation. Secondly, we showed using our head pose evaluation database, that tracking the head location and pose jointly is more efficient than first tracking the location of the head, and then estimates its pose. Lastly, by comparing the performance of three SMC based techniques for joint head location and pose tracking, we showed that the Rao-Blackwellized method provides more stable results overall.

1.2.3 VFOA recognition in meetings

Our tracking systems output head poses that are used for VFOA modeling. Using the meeting room recordings of our head pose database, we investigated the modeling of the VFOA of people involved in a 4-persons meeting situations. For each person, the set of possible VFOAs was discretized into 5 targets: the 3 other participants, the table, the projection screen and a target denoted unfocused when the person was not focusing on any of the previous VFOA targets. Figure 1.3 shows the geometric set up of the meeting room where our study took place, and Figure 1.3(a) shows the VFOA targets for one of the person in our set up. Our set up is more complex than previous works on VFOA recognition where the number of VFOA targets were 3 and the specific VFOA could be identified from the head pan angles only. In our set up, for each person there is 6 potential VFOA targets. As VFOA model, we investigated the use of both a GMM and an HMM model where the hidden states are the VFOA targets we want to recognize and the observations are the head poses obtained, from the magnetic field sensor or a vision based tracking system. Using the head pose ground truth from the magnetic field sensor allowed us to compare the optimal VFOA modeling situation where head pose ground truth are available in comparison to the situation where only head pose tracking estimates are available.

To learn the parameters of the emissions probability models (Gaussian) for the observations, two methods were investigated. A first method is based on machine learning where, part of the head pose data

Figure 1.4: Moving people's VFOA recognition. People passing by a glass windows are tracked, their head poses estimated. A model using as observation head locations and poses is used to estimate whether or not they are focusing an advertisement posted on the glass windows.

are used to train the VFOA models, and the remaining part is used to test the models. A second method relies on the geometry of the meeting room, as well as on the fact that gazing at a target is accomplished by both a head rotation and a eye-in-head rotation. Also, because people have personal ways to focus at targets, unsupervised maximum a posteriori adaptation (MAP) of the generic VFOA models is applied to the input test data. Thorough experimentation is conducted to evaluate our models. In summary, the contributions of our investigations about static people VFOA recognition are the following. First, we study VFOA recognition for in a realistic and complex meeting situations where each person has potentially 6 possible VFOA targets. Secondly, we proposed a model to derive the VFOA of a person from his head pose. The proposed model exploits knowledge about the meeting room geometry and people/target locations. Thirdly, we proposed the use of an unsupervised MAP adaptation framework to adapt the VFOA model parameters to individual people. Finally, we proposed an event-based VFOA performance measure, used together with the classical frame-based measure, and used the performance measures in a thorough experimental study of VFOA recognition.

1.2.4 VFOA recognition for moving people

In this thesis, we also investigate the recognition of the VFOA of moving people. The task considered in this work is, given an advertisement posted on a glass window, to recognize whether or not people passing by were focusing or not on the advertisement. Figure 1.4 illustrates our scenario. We proposed a probabilistic multi-person tracking model based on reversible jump Markov Chain Monte Carlo (RJ-MCMC). The state to estimate is a multi-person models encoding information about the person's body, location, head location and pose. RJ-MCMC allows to estimate the pdf of a state with varying dimensionality. RJ-MCMC is well suited because the number of people in the scene is varying and MCMC is more efficient in high dimension space than standard SMC techniques such as the ISPF method. The multi-person tracking models outputs the head location and pose together with the person's identity. The head locations and poses are used as observations in a GMM framework to model people focusing the advertisement. The contributions of our investigation about the VFOA recognition for moving people are the three-fold. First, we define a new problem, namely, the VFOA estimation of moving people, which generalizes previous studies about the VFOA estimation for static persons. Secondly, we generalized previous work about multi-object tracking using RJ-MCMC with a state space defined on people's body location, head location, and pose. Thirdly, we proposed a VFOA model for moving people. Lastly, we demonstrated the applicability of our model by applying it to an outdoor advertisement problem.

1.2.5 Conclusion

Because of its importance in the understanding of human behaviors by computers, the present dissertation reports on the developementand evaluation of advanced methods toward the automatic inference of people's VFOA from their head poses. In **Chapter 2**, we review background works about the main topics of this thesis, namely head modeling, tracking and VFOA recognition. **Chapter 3** describes the public head pose video database we built for head tracking and pose evaluation using a 3D location and orientation tracker. **Chapter 4** describes head pose tracking methodologies. In **Chapter 5**, we present our studies about VFOA modeling of static persons in meeting situations. **Chapter 6** describes one approach to recognize the VFOA for moving people. **Chapter 7** concludes the thesis by discussing our work and conjecturing possible future investigations.

Chapter 2

Background and related work

The study of the visual focus of attention (VFOA) usually required the eye gaze. In absence of gaze information due to low resolution imagery, the head pose is a good indicator of the VFOA. Studying the head pose requires head [1] modeling and tracking. This section reviews existing research related to the central topics of this thesis, namely head modeling, object tracking and VFOA studies. Section 2.1 is devoted to head modeling, Section 2.2 to single object tracking. Section 2.3 is devoted to the analysis of the role of the VFOA from a social interaction point of view. Studies related to the computer vision aspect the VFOA will be discussed in Section 5.

2.1 Head modeling

Head modeling in computer vision is a widely investigated topic. Modeling head is necessary for face detection, face verification, face recognition, head tracking, facial expression recognition and human avatar building. Head modeling is a difficult task because, in real life, head appearance is subject to variation depending on various conditions:

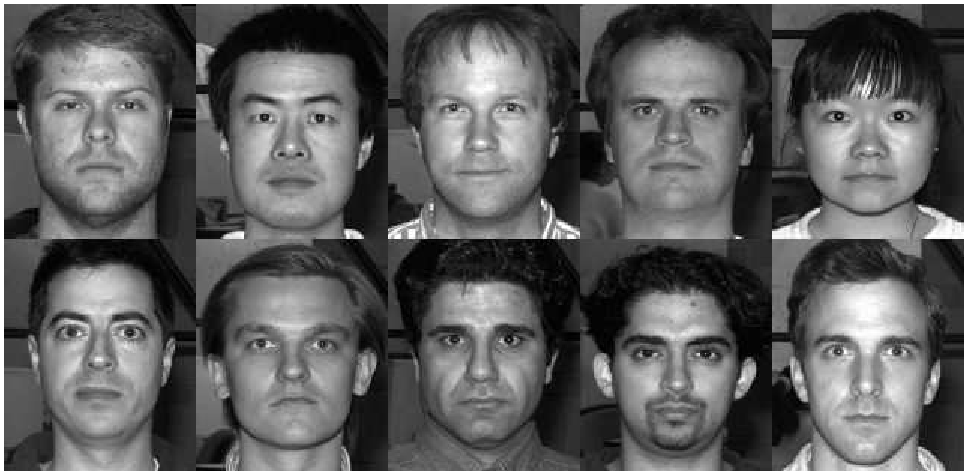
- **Scale variation:** when a person is moving closer or farther to the camera his head is subject to size changes.

- **Head orientation:** obviously, a head does not look the same from in all it's sides. For instance, in one side is the face, on another is the back of the head. When a person changes his orientation with respect to the camera, his head appearance changes depending on the visible part of the head.

- **Illumination conditions:** depending on the illumination sources locations, head appearance is subject to variation because the different parts of the head surface do not reflect the light in the same way, and reflection depends on the unknown geometry (light sources, camera, objects). Thus, illumination vary locally or globally. Each illumination conditions induces specific head appearance variations as can be seen in 2.1(a).

[1] In the text we will not make difference between models using face and hair and models using the face only. All of them will be called head models

(a) Head pose and illumination appearance variations (images from the CMUPIE Database).



(b) Head appearance variations across individuals (images from the Yale Face Database).

(c) Head occlusions (image from AR Database).

Figure 2.1: Head appearance variations.

- **Intra-person variability:** for the same person, head appearance is subject to variations. People do not have every time the same hair cut. Sometimes people have facial hairs (beard, moustaches) some other time not. People also use make-ups to modify their face appearance.

- **Extra-person variability:** different person, in general, have different head appearances (see Figure 2.1(b)). Head anthropometric characteristics are person-dependent. People have different eyes, nose, mouth sizes. Some people have facial hairs, long or short hairs, other people not.

- **Occlusions:** head appearance changes when it is occluded by another object as can be seen in 2.1(c). People frequently have their head partially occluded by their hands, hats or glasses.

- **Feature space dimensionality:** in the case of a head patch image of 32×32 pixels, which is a small size, the head patch belongs to a 1024-dimension space. Modeling in such dimension is difficult because, in general, distributions are peaky and multi-modal.

Efficient head modeling should take into account most of the difficulties cited above. Depending on the goal, several classes of methods have been proposed:

Head detection: In a head detection task, the head location is unknown. The goal is to build models that can be used to detect heads in images. This situation requires models that are able to discriminate a head from what is not a head, called non-head. Detection gives rise to the group of methods called discriminative models. Discriminative models are efficient in localizing the heads but they will fail in identifying precise qualitative aspect of the head. For example, discriminative methods will be able to localize the head without being able to give precise information about the person's identity. An important aspect of head detection is the head search. Because, in general, there is no prior information about the head locations, efficient search strategies have to de defined.

Head tracking: In head tracking, the task is, given the head location at a given time, to infer the head location at the next time. In this situation it is only required to the models to be find the head by a local search. In a tracking situation, the head search is less a critical problem than in head detection. The head search is done in a local area around the previous head location. Although discriminative models can also be used, another set of models called the generative model can also be used. The generative models are not in general able to differentiate head from non-head, but than can successfully track the head using the previous head location as prior knowledge.

Face verification: In face verification, the head location is available. The task is to identify a qualitative aspect the head such as a person identity. In a face verification situation, it is not required that the head model is able to differentiate a head from a non-head but discriminate between heads of different persons. In this situation, generative head models are used. Since in this case few training data, per identity, are available and model adaptation is well defined in a generative model framework.

In the following, we will first describe the features used to model heads, and will then present a set of head modeling approaches: the template models, the eigenspace based models, the discriminative based representation, and the generative models.

(a) Image sample. (b) Detected skin color areas using a Gaussian skin color model in the normalized RG space. (c) Edges detected by thresholding the gradient's norm.

Figure 2.2: Example image and it corresponding skin color and edges features.

2.1.1 Head modeling features

Head modeling is done using image features. The features used to model head are mainly skin color, edges and textures.

Skin color features

Skin color has proven to be an effective cue to model human skin. It can be used to localized human head from the face skin color as shown in Figure 2.2(b). Although the skin color range is wide, from black to white people, several studies have shown that the differences between the skin pixels values lie largely between the intensities and not the color. Many color spaces have been used to model skin pixels: Red-Green-Blue (RGB), normalized RGB, Hue-Saturation Value (HSV), YCrCb, CIE LUV. Two popular methods to model skin color exist. Non parametric models such as histogram of skin pixel values [2] and parametric models such as Gaussian mixture model (GMM) fitted to skin pixel values [3]. [4] presents a thorough comparison of various skin color segmentation techniques.

Computational fastness is the main advantage of modeling heads using skin color. There are two main drawbacks of skin color modeling. First, skin color like object are very common in real life. In Figure 2.2(b), it can be seen that the hand of the person will be a serious ambiguity when trying to localize the head using skin color. Secondly, skin color models are sensitive to illumination conditions. Some color space such as HSV or normalized RGB are designed to be robust to illumination variations. Another solution to be robust to illumination changes is to use model adaptation such as proposed in [5]. When skin color is modeled in a GMM framework, maximum a posteriori (MAP) adaptation can be used to be more robust to illumination changes.

Edge features

An edge in an image can be defined as a connected sets of discontinuity pixels. Human heads are characterized by typical edges distributions. Thus, edges can be used as features to model heads. A simple method to detect edges in an image is to threshold the norm of the gradient of the image (see Figure 2.2(c)). Discontinuity pixels are pixels where the norm of the gradient is high. More sophisticated methods to detect edges in images, such as the Canny edge detector or the Sobel edge detector, exist in the literature [6].

The major drawbacks of edge features are, first, edge detection require threshold that are in general image dependent. Illumination conditions can change edge distributions. A solution to make edge extraction and binarization more stable is to normalize the image with respect to illumination using techniques such as histogram equalization. A second limitation of edge features is their sensitiveness to textured background clutter. In a textured background the presence of many edges increase the possibility of false alarm. A solution to this issue is to use edge features with skin color features.

Texture features

Broadly speaking, head images are characterized by specific texture maps. Thus, texture can be the basis of head models. The simplest texture map is the raw gray levels image template. Although sensitive to illumination conditions, raw gray level intensities have been successfully used for head modeling. More sophisticated texture maps can be defined as multi-scale differential image information obtained by filtering image with gradient operators such as multi-scale derivative of Gaussian (DoG), Laplacian of Gaussian (LoG). Wavelets also have been used to obtain texture maps [7]. The main advantage of wavelets is their ability to capture edge related information at specified scales and orientations. Recently, the local binary pattern (LBP), a new representation of texture maps has been proposed [8]. For each pixel, the LBP encodes if its neighboring pixels have higher or lower intensity values. LBP is robust to illumination changes because it encodes only relative relationship between pixels that are still valid for a large range of illumination conditions. Another texture representation, the scale invariant feature transform (SIFT), is a representation used to capture oriented texture information around a point [9]. One of the advantages of the SIFT representation, is that it specifies the best scale at which the texture features should be represented.

Texture are subject to similar drawbacks than edge features. Similar solutions such as combining texture with skin color to be more robust to background clutter or normalizing the image with respect to illumination conditions can be used to handle these drawbacks.

In the present thesis we will show that the joint use of skin color, edge and texture information can yield robust head models. In the following we thus describe various approaches that have been used for head modeling in computer vision.

2.1.2 Template models

A simple way to model a head is to use a template selected by an “expert”. An example of head template model is a patch image of a person’s head extracted from an image. Head search in images can be done by finding the configurations maximizing a similarity criteria with the template. In addition, head can be represented by a set of more specific patterns. For instance, [10] proposed for tracking to represent a head by a template composed by five elements: 2 image-patches for the eyes, 2 image-patches for the eye brows and one for the nose. Besides the requirement to be expert-selected, templates have some drawbacks. Templates badly generalize to other head appearances because they are built from a single example. Templates also badly generalize to other experimental conditions such as other illumination conditions. A more sophisticated template modeling is the multi-dimensional morphable model proposed in [11]. In the multi-dimensional morphable model framework, a face image, up to a warping in scale, is

(a) Eigenfaces: mean (top left) and 19 first eigenfaces trained from a set of frontal head images.

(b) Head reconstruction from eigenfaces. First row input faces, second rows reconstructed faces using the eigenfaces in Figure 2.3(a).

Figure 2.3: Eigenfaces and reconstruction examples.

taken to be a linear combination of a set of face templates. A linear combination of multiple templates has the advantage to be more robust to appearance and illumination changes.

Templates have also been used to for head orientation recognition [12]. In [12] a set of head templates from various persons at various orientations, organized in a tree-structure, was used for head pose recognition. The pose of a new head image is recognized by finding in the tree the most similar template to the input head image.

The previous described template models are based on the pixel intensities. Another set of template models are based on edges. A simple edge based template is to represent a head by it's surrounding contour, which at the simplest case, can be assumed to be an ellipse. Since the use of an ellipse template for head tracking in [13], head shape models have been very popular in the head tracking community [14, 15]. More sophisticated shape template models, defined by set of landmarks around the face and the facial features (nostrils, eye, nose mouth), can be obtained in an active contours framework [16], or using the elastic bunch graph matching framework [17], or in an active shape models (ASM) framework [18].

The ASMs, presented in [18], are based on principal component analysis (PCA). The principle of ASMs is the following. We assumed available a set of training head images with annotated landmarks defining the head shapes, such as showed in Figure 2.4. Using PCA, a statistical model s_m of the head shape s is learned as

$$s = s_m + e_s \text{ with } s_m = \bar{s} + P_s b_s \tag{2.1}$$

where $\bar{s}$ denotes the mean shape, P_s is a matrix representing a set of orthogonal modes, b_s is a set of

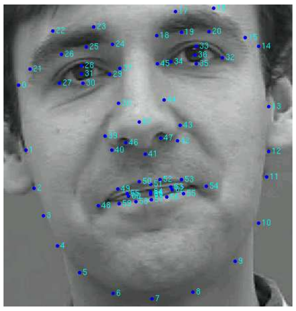


Figure 2.4: Face with annotated landmarks for ASMs and AAMs training.

shape parameters, and e_s is the residual error for the shape approximation. The major drawback of ASMs models, and at the same time of the other shape models, is that they only model geometric head shape variations based on the edges present in images. Thus, shape models are very sensitive to textured background. However, ASMs are interesting because they can be used for head pose modeling. The only requirement is to include in the training data, head pose variations similar to the one that needed to be modeled. Although, the pose variations have to be restricted to near frontal head poses because, close to near profile poses, the edges defining the head boundaries are not visible anymore.

2.1.3 Eigenspace models

Closely related to template modeling is the eigenface representation [19]. An eigenface is the representation of a head using PCA. An eigenface is obtained by decomposing a set of training head images using PCA and keeping only the eigenvectors with highest eigenvalues (see Figure 2.3(a)). A new test image is then classified as a head or non-head depending on it's reconstruction errors. The eigenface representation, by discarding the eigenvectors with the lowest eigenvalues, keeps only the common characteristics of the training images and removes the other characteristics which are considered to be noise. In Figure 2.3(b) the removal of the characteristics considered as noise results in reconstructed images that more blurred than the input images.

More generally than in [19], where only frontal faces were modeled, the eigenface representation can be used to model a head seen from various orientations. This was done in [20] where the space of head pose was discretized, and for each discrete head pose eigenfaces are built. The pose of a new head image is then estimated as the pose of the eigenface models which allows for the best reconstruction of the input image. Another method to represent an object seen from many orientations using eigenvectors was proposed in [21]. The method represents an object by decomposing example images of the object seen at various orientations using the eigenspace representation. Similar representation was used by [22] to represent faces. Figure 2.5 shows example of eigenfaces in this framework. Figure 2.6 shows reconstructed heads using the framework proposed in [22]. The number of eigenfaces required to obtained a good reconstruction, approximately 200, is a lot higher when the head is seen at multiple orientations than when the head is seen from a single orientation, where 20 eigenvectors are sufficient. The disadvantage of retaining a high number of eigenfaces is that, since an input head image has to be projected on the all eigenfaces, the computational cost of the projection becomes important. It has to be noticed that

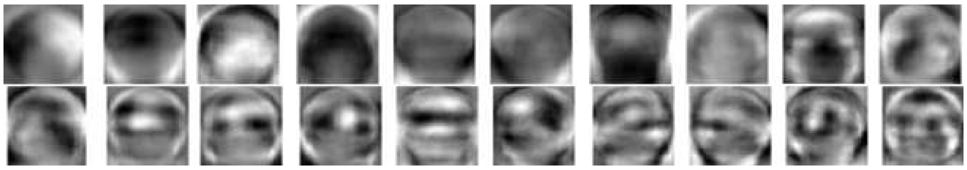
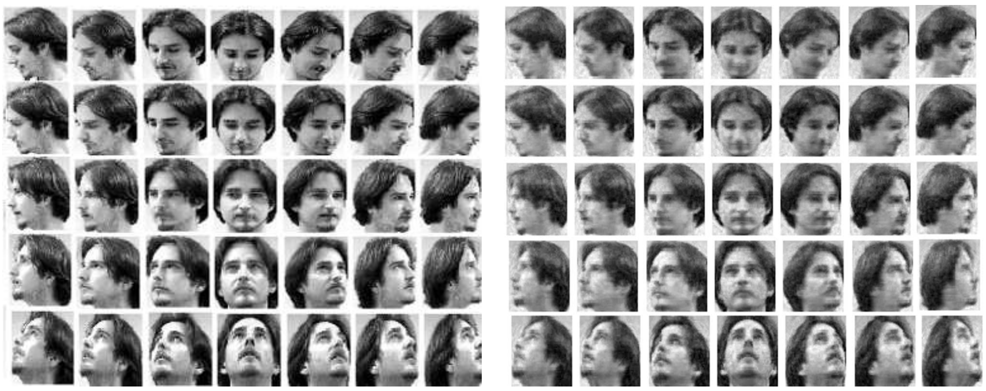


Figure 2.5: Eigenfaces: 20 first eigenfaces, using a training set of head images with different head orientations.



(a) Input heads.

(b) Reconstructed heads using the 200 first eigenfaces.

Figure 2.6: Head reconstruction from eigenfaces learned with various orientations head appearance.

the main difference between the representation in [22] applying PCA to appearances at various orientations and the representation in [20] applying PCA at a single orientation lies in the training data. In both works, a similar PCA framework is applied but to different data set.

More generally, head seen at various orientation are characterized by different edges maps. The eigenfaces can be seen as a basis oriented edge detection operators [22]. Figure 2.5 shows examples of such eigenfaces. To generalize the eigenface representation which consists in a linear decomposition of the input feature space, other researchers proposed to use a non linear decomposition of the input feature space on a Gabor wavelet network (GWN). GWN are a set basis vectors capable of capturing edges at various scales and orientations [23]. Eigenspace representation can also be seen a dimensionality reduction technique based on a linear decomposition of the input feature space. The eigenspace dimension is, in general, lower than the input feature space. Other authors proposed to apply non-linear dimensionality reduction techniques such as locally linear embedding (LLE), or ISOMAPS to the head pose modeling problem [24, 25, 26].

2.1.4 Discriminative models

Initially, discriminative methods were used for object detection. Nowadays, they are also used for tracking. In discriminative approaches, the goal is to build discriminative functions separating in the feature space the head and non-head classes. Nonlinear decision surfaces can be formed using neural networks or support vector machines (SVMs).

In the early work on face detection some authors proposed a 3 layers neural network model [27]. The first layer receives the input gray level image resized to a dimension of 20×20. The second layer comprised 3 types of hidden units: 4 units looking at 10×10 pixel subregions, 16 units looking at 5×5 pixels subregions, and 6 looking at 20×5 horizontal stripes. Each of the units was chosen to represent localized features (eye, nose, mouth) important for face detection. The output layer gives the confidence in the presence of absence of a frontal face in the input patch image. Other neural network structures have been proposed for head modeling such as a linear auto-associative memories (LAAM) to classify face by gender [28]. Linear auto-associative memories are particular cases of single layer linear neural network where input patterns are associated with each other. Another way to model head in a discriminative fashion is to use the SVM framework. Contrarily to neural networks which minimize training errors, making generalization to new appearances more problematic, SVMs minimize an upper bound of the generalization error through the use of the margin concept. The applicability of SVM for head modeling in a frontal face detection problem have been confirmed in [29]. The authors in [30] proposed to use AdaBoost [31] to combine a set of simple classifiers to form a stronger classifier for frontal face detection. One of the main contribution of this work was the combination of classifiers in a cascade allowing to quickly discard in the search for face non-informative image regions and focus on face like objects.

Discriminative models can also be used to model heads seen from various orientations. Most of the discriminative head pose models are generalization of discriminative near frontal head models. The frontal face detection method proposed in [30] based on AdaBoost was generalized to detect faces at various orientations [32, 33]. In [34], a 3 layers neural network was proposed to model head at various orientation. The first layer receive the intensities of the gray level image. The third layer output confidence of the input head image being at a given orientation. Other neural network structures, such as LAAMs or multi-layer peceptrons, have also been used for head pose modeling. In [35], the head pose

(a) Gaussian means. (b) Gaussian standard deviations.

Figure 2.7: Head orientation modeling using Gaussian distributions.

space was discretized into finite values and, for each discrete pose value, a LAAM was used to model head appearances. A multi layer perceptron has been used in [36] to model head poses using Gabor features as input instead of gray level pixel intensities. The Gabor features had the advantage to be more robust to illumination changes and to contain information about contrast, spatial information, scale and orientation. SVMs also have been used to model head poses. In [37], it was proposed to use 2 SVMs, having as input face image PCA features, to estimate a face pose. Similar SVM modeling for head orientation tracking was proposed in [38]. Finally, another interesting discriminative head pose modeling approach includes the works in [39], which is based on a non parametric approach. In this work, local head appearances were discretized into a given number of patterns. An estimate of the probability of a face presence, given the discrete patterns, was computed in closed form by counting the frequency of occurrence of patterns over training image sets.

2.1.5 Generative models

In head modeling, the word "generative" is usually used to designate two terminologies. First, in a statistical framework, it refers to approaches that rely on probability distributions to represent head classes. Secondly, it also refers to approaches that aim at synthesizing head appearances. Among generative approaches to model heads we can cite, distribution based models, active appearance models (AAMs), and 3D-based head models.

Distribution based models

Distribution based approaches for head modeling can be posed in a probabilistic framework. A feature vector z computed from an image patch is viewed as a random variable. This random variable is related to the presence or absence of a head by the class conditional distributions $p(z|\text{head})$ and $p(z|\text{non head})$. Given the class conditional distributions, Bayesian classification can be used to classify an image as head or non-head. In a head detection problem, both of the distributions $p(z|\text{head})$ and $p(z|\text{non head})$ are used. But, in a tracking situation, only $p(z|\text{head})$ is required.

Demonstration of the usability of distribution based models for face detection was made in [40]. This work proposed a distribution based method for frontal faces detection which showed how the distribution of the patterns from one class $p(z|\text{head})$ can be learned from positive examples. The distribution $p(z|\text{non head})$ was learned from negative examples (non head patterns). Using as feature vector z the pixel intensities of gray level face images resized to a reference size, the technique models the distribution of the observation for human faces $p(z|\text{face})$ by six clusters. Each cluster is represented by a multi-dimensional Gaussian distribution with mean and covariance matrix obtained by clustering the face training features into six clusters using k-means algorithm. Similarly, six non-face clusters, learned from non head patterns, are used to model the non-head class.

A visual learning method based on density estimation in high dimensional space using PCA was proposed in [41]. PCA is used to define the subspace that best represents a set of patterns by decomposing the feature vector space into mutually exclusive and complementary subspaces. A multi-variate Gaussian mixture model is used to learn the statistics of the local features of a frontal face. In the framework presented in [41], only face pattern distributions were learned. Because modeling non face patterns distribution improves head detection, others proposed a similar approach that includes the modeling of the distribution of the non-face patterns [42].

Distribution based-models can be a framework for head pose modeling. For instance [43] presented a method to detect frontal, left and right profile faces for any size and location. The head parts (eye, nose, mouth) features class conditional probabilities $p(z_{part}|\text{face})$ and $p(z_{part}|\text{ non face})$ are modeled as look up tables instead of continuous distributions. Representing the distributions as non-parametric tables avoid to arbitrarily define their nature. In [43], features were obtained by applying wavelet transform to grey level images allowing to capture edge structures characterizing heads.

Other distribution based head pose modeling include [44] who proposed for head pose tracking, to model a head in the 3-dimension (3D) space as an ellipsoid with a set of point on its surface. Each point, indexed by i, is represented by its coordinates and a Gaussian probability distribution function $p_i(z|\theta)$ representing the belief that given a particular head pose, the point i will project observation z. Several texture based observations were compared in [44], including Gabor wavelets features, a Gaussian at coarse and rotation invariant Gabor wavelets, a Gaussian and Laplacian features. The best head pose representation were obtained with a Gaussian at coarse and rotation invariant Gabor wavelets. Showing that even more sensitive to illumination conditions, information from the gray level pixels intensities are not negligible for head orientation estimation.

Using the best features found in [44] as observation, other researchers proposed for head orientation modeling to discretize the head pose space into a discrete set θ_k [45]. The class conditional probability distributions of the observations given a head orientation $p(z|\theta_k)$ were represented in 2D as Gaussian distribution, such as illustrated in 2.7. Although this representation decreases orientation estimation accuracy with respect to the 3D representation presented in [44], because of the discretization, it has the advantage that discretized head pose databases exist already for training the models [46, 47, 48].

Active appearance models (AAM)

The AAMs introduced in [49] are a generalization of the ASM [18]. An AAM allows to represent head appearance variations due to identity, orientation and facial expressions variability. Thus, AAMs can be used to synthesize heads. The learning of AAMs requires a set of labeled head images with a set of control points defining the head shape as shown in Figure 2.4. A statistical model, s_m, of the head shape,

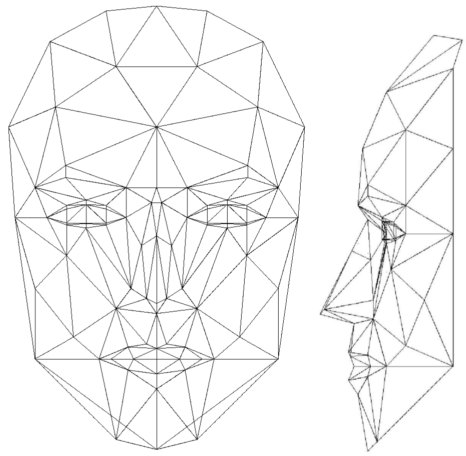


Figure 2.8: CANDIDE-3: a 3D face model (from [55]).

s, can be learned using the ASM framework as

$$s = s_m + e_s \text{ with } s_m = \bar{s} + P_s b_s, \tag{2.2}$$

where $\bar{s}$ denotes the mean shape, P_s is a matrix representing a set of orthogonal modes, b_s is a set of shape parameters, and e_s is the residual error for the shape approximation. To build a statistical model of the grey level appearance, each example image is warped so that its control point match the mean shape $\bar{s}$. Then, the pixels intensities inside the mean shape of the warped example image are sampled. By applying PCA to the normalized data, a linear face appearance model is obtained as

$$a = a_s + e_a \text{ with } a_s = \bar{a} + P_a b_a, \tag{2.3}$$

where $\bar{a}$ is the mean normalized gray level, P_a is a set of orthogonal mode variation, and b_a is a set of grey level parameters allowing to synthesize the appearance a. The appearance approximation error is denoted e_a. The ability of an AAM to model a head at various orientations depends on the data used to learn the models. If in the training data there are variabilities with respect to the head orientation, the AAM will be able to model those variations. AAMs have been successfully used in many researches involving head modeling such as face recognition [50], face and facial features tracking [51, 52].

Recently, the AAM theory have been revisited in [53] which proposed new schemes for a more efficient fitting of AAMs to input test images. [54] proposed the generalization of AAMs to more complex illumination conditions using image-based rendering techniques which can represent complex lighting conditions, structures, and surfaces. The major drawback of AAMs is that, because they require the visibility of the face shape boundaries, modeling extreme head orientations such as head profiles is not well defined. Also, because AAMs modeling rely on the facial features, it is required, for a good fitting, that head images are of high resolution so that the facial features are visible enough.

3D model based approach

A head can be modeled in 3D using a parameterized mask. The CANDIDE model proposed in [56] is a parameterized face developed for model based coding of human face. It is composed of vertices (100) and triangles (75) defining the face surface. Face motions are controlled by action units (AUs). Recent version of the CANDIDE model contains more vertices and triangles for a better facial action modeling [55, 57]. The CANDIDE model can be seen as a $3N$ dimensional vector $\bar{v}$, where N is the number of vertices, containing the 3D coordinates of the vertices. The model is reshaped according to

$$v(\varsigma, \vartheta) = \bar{v} + S\varsigma + A\vartheta \tag{2.4}$$

where the vector v contains the new vertex coordinates, and S and A are shape and animation units. Thus ς and ϑ are shape and animation parameters defining the AUs. The face texture for each triangle is extracted by fitting the model to an initial face image of known orientation (frontal in general). To synthesize a global motions, few parameters controlling rotation, scaling and translation are added to the model

$$v(\varsigma, \vartheta) = Rs(\bar{v} + S\varsigma + A\vartheta) + t \tag{2.5}$$

where R is a rotation matrix, s is a scale factor, and t is a translation vector. Thus, 3D head models are head pose models, because they include in their definition variation with respect to head rotations.

Other works, [58, 59], use face models similar to the CANDIDE model. Other CANDIDE-like-models include in the modeling people's hair and integrate model adaptation framework [60]. Also closely related to the CANDIDE model is the 3D ellipsoidal surface head model proposed by [61].

By definition 3D head models are well suited to study facial feature actions. Depending on the precision of the vertexes defining the model, the CANDIDE can model very precisely the motions of each part of a human face. But, the 3D models require high resolution imagery. The main reason is that the vertexes defining the models are built with respect to the facial features. Localizing and tracking these facial features are reliable only on high resolution images.

2.2 Object tracking

Heuristically speaking, tracking an object through a sequence of images consists of estimating a parametric representation of the object. At each time, the previous time parametric representation of the object is usually supposed to be known and used as prior knowledge. Otherwise it is an object detection problem. In a simple case, the parametric representation of the object X, called the object state, can be a bounding box, parameterized by a center point, a width, and a height, locating the object in the image plane (see Figure 2.9).

Many approaches have been adopted to tackle the tracking problem. Among them we can cite, the bottom-up and top-down approaches. Bottom up approaches tend to construct object state by analyzing the content of images. Segmentation based methods such as optical flow based tracking are bottom-up approaches. On the contrary, top-down approaches tend to generates state hypotheses from previous time based on a parametric modeling of the object. Tracking is done by verifying the predictions on image

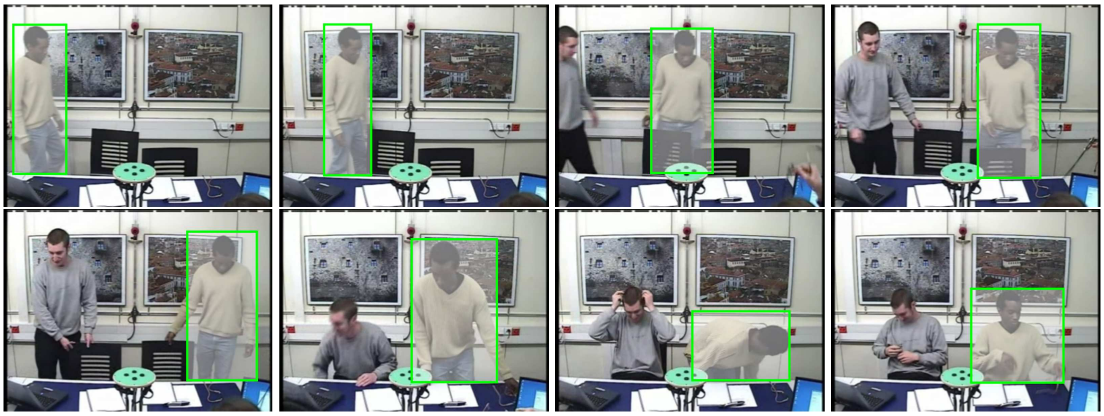


Figure 2.9: Object tracking illustration: a person represented by a bounding box, parameterized by a center point and a width and height, tracked through a sequence of images.

measurements. Another important characteristic of tracking approaches is whether or not they are based on a Bayesian framework. This is the topic of the following Section.

2.2.1 Non-Bayesian tracking

Non-Bayesian tracking methods do not use a Bayesian formalism to solve the tracking problem. Among the non-Bayesian tracking methods we briefly discuss below the optical flow based tracking and gradient-descent based tracking such as the mean-shift algorithm.

Optical flow tracking

Optical flow estimation can be used for tracking by segmenting regions with similar motion fields when tracking rigid objects [62, 63, 64]. The principle of this optical flow is to estimate dense motion fields from spatio-temporal patterns based on the assumption that $I(x,t)$, the image brightness at a pixel location x for a given time instant t, is constant. Therefore, the derivative of the image brightness with respect to time is null, ie.:

$$\frac{d}{dt}I(x,t) = 0. \tag{2.6}$$

The optical flow estimation problem is thus to estimate the motion field $v = \frac{\partial x}{\partial t}$ such that:

$$\nabla I.v + \frac{\partial I}{\partial t} = 0. \tag{2.7}$$

However, this single equation is not sufficient to recover the flow field. Various solutions have been proposed to solve this problem by adding other constraints [65].

There are some drawbacks related to the optical flow estimation problem making tracking using optical flow sometimes uncertain. First, it relies on the assumption that the image brightness at pixel locations are constant which is not always the case. Images brightness are subject to changes that are not only due to motion. For example, brightness change can be due to illumination conditions. Secondly, optical flow estimation becomes intractable when tracking possibly rotating large un-textured objects with all pixels with a similar brightness.

Gradient descent based tracking

Tracking can be posed as a minimization problem. Let us denote by z_{obj} a model of the object in the observation space, z_X the observations extracted from the candidate state configuration X, and ρ a distance in the observation space. Tracking can be posed as an optimization problem. For instance, tracking can be posed as finding an optimal state configuration X_{opt} verifying

$$X_{opt} = \arg\min_X \rho(z_{obj}, z_X). \tag{2.8}$$

In the cases the distance ρ and the observation function $h(X) = z_X$ admit good differentiability and convexity properties, the optimal state configuration can be obtained using gradient descent. In a tracking formulation, the initial state of the optimization process is taken to be the optimal state of the previous time or a prediction made out of it.

The mean shift algorithm for tracking, proposed in [66], is a gradient descent based tracking technique. In the mean shift framework, the target of interest is represented by a kernel density distribution z_{obj}. At a candidate object configuration, the object is also represented by a kernel density distribution z_X, and the distance in the observation space is the Battacharya distance, proposed in [67], defined as:

$$\rho(z_{obj}, z_X) = \sqrt{1 - \int_u \sqrt{z_{obj}(u) z_X(u)} du} \tag{2.9}$$

Gradient descent based techniques have the advantages to be very fast techniques because the framework provides an optimal search direction. Algorithms such as the steepest descent method are guaranteed to converge after few iterations. Fastness of convergence is important for people designing real time systems. A major drawback of gradient-descent based techniques is that they converge to the local minima closest to the initialization point. They are easily distracted by ambiguities. Thus, gradient-descent based techniques do not cope very well with multi-modal distributions because they track only a single mode. Also, because in tracking an analytical expressions of the observation function $h(X)$ is not always available, the requirement of a differentiable objective function is not always full-filled. When good objective function properties are not full-filled, solutions such as template matching can be adopted with ad-hoc search strategies. But, depending on the state dimensionality the search space can be very wide making the search matching computationally very expensive.

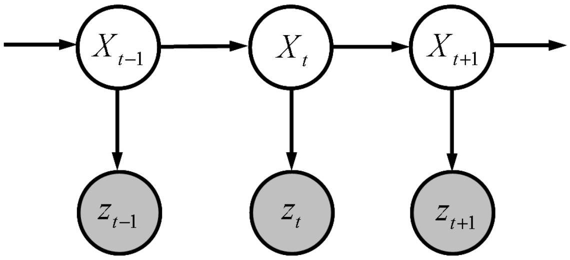


Figure 2.10: Standard graphical model for tracking.

2.2.2 Bayesian tracking

Problem formulation

Bayesian tracking can be stated in a general formulation. To define the Bayesian tracking problem, we consider that the evolution of the state sequence $\{X_t,\ t \in \mathbb{N}\}$ of an object is given by:

$$X_t = f_t(X_{t-1}, \eta_{t-1}) \tag{2.10}$$

where $f_t : \mathbb{R}^{n_X} \times \mathbb{R}^{n_\eta} \rightarrow \mathbb{R}^{n_X}$ is possibly a nonlinear function of the state X_{t-1}, and of η_t where $\{\eta_t\ t \in \mathbb{N}\}$ is an independent identically distributed (i.i.d.) process noise sequence. n_X and n_η are respectively the dimensions of the state and the process noise vector. The objective of tracking is to recursively estimate the state X_t from the measurements z_t:

$$z_t = h_t(X_t, \nu_t) \tag{2.11}$$

where $h_t : \mathbb{R}^{n_X} \times \mathbb{R}^{n_\nu} \rightarrow \mathbb{R}^{n_z}$ is a function, possibly nonlinear, of the state X_t and of ν_t, $\{\nu_t\ t \in \mathbb{N}\}$ is an i.i.d. measurement noise sequence. n_z and n_ν are respectively the dimensions of the measurements and noise vectors . In particular we seek for estimates of X_t based on the set of all available measurements $z_{1:t} = \{z_l,\ l = 1, ..., t\}$ up to time t.

Bayesian tracking thus requires the definition of three models:

1. A state model X which is a parametric representation of the object.
2. A dynamic model, represented in Equation 2.10 by the function f_t, which parameterizes the temporal evolution of the state.
3. An observation model, represented in Equation 2.11 by the function h_t, which measures the evidence that can be observed in images about the state.

From a Bayesian perspective, tracking consists in recursively estimating some degree of belief in the state X_t of the target given the sequence of observations $z_{1:t}$. Solving the Bayesian tracking problem requires to construct the conditional probability density function (pdf) of the states given the data, $p(X_t|z_{1:t})$. The distribution $p(X_t|z_{1:t})$ is also known as the filtering distribution. If we assume that the initial pdf $p(X_0)$ is known, the pdf may be obtained in two stages: prediction and update. Suppose that the pdf at time $t-1$, $p(X_{t-1}|z_{1:t-1})$, is available. The prediction stage involves the use of the system model in Equation 2.10 to obtain the predicted pdf of the state at time t via the Chapman-Kolmogorov equation:

$$p(X_t|z_{1:t-1}) = \int_{X_{t-1}} p(X_t|X_{t-1}, z_{1:t-1})p(X_{t-1}|z_{1:t-1})dX_{t-1}. \tag{2.12}$$

The dynamical model in (2.12), $p(X_t|X_{t-1}, z_{1:t-1})$, models the probability to be in a state X_t given the state at the previous time, X_{t-1}, and the sequence of observations, $z_{1:t-1}$. In general the state X_t is assumed Markovian. In that case, the dynamical model can be written

$$p(X_t|X_{t-1}, z_{1:t-1}) = p(X_t|X_{t-1}) \tag{2.13}$$

leading to the graphical model in Figure 2.10. In case the evolution state function in Equation 2.10 is a linear function defined as

$$X_t = f_t(X_{t-1}, \eta_{t-1}) = F_t X_{t-1} + \eta_{t-1}, \tag{2.14}$$

in the Markovian case, the state dynamical model can be written

$$p(X_t|X_{t-1}) = \mathcal{N}(X_t, F_t X_{t-1}, \Sigma_{t-1}). \tag{2.15}$$

where Σ_t is the covariance matrix of the state noise process η_t. More precisely, the state dynamical model is a Gaussian distribution with mean $F_t X_{t-1}$, and with covariance matrix Σ_{t-1}.

At time t, the measurement z_t is used to update the predicted pdf via Bayes' rule:

$$p(X_t|z_{1:t}) = \frac{p(z_t|X_t, z_{1:t-1})p(X_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{2.16}$$

The term $p(z_t|X_t, z_{1:t-1})$ in (2.16), called the object likelihood, measures how well the new observation is in adequacy with the hypothesized state given the past observations. $p(X_t|z_{1:t-1})$ is the prediction distribution given in Equation 2.12. $p(z_t|z_{1:t-1})$ is a normalization constant independent of the state. Because the task is to estimate the state, knowing exactly the normalization constant is not necessary to solve the tracking problem.

The recurrence relations in (2.12) and (2.16) form the basic of the Bayesian tracking formalism. Depending on the hypotheses on the state model in (2.10), and on the observation model in (2.11), or more generally depending on the modeling of the distributions $p(X_t|X_t, z_{1:t})$ and $p(z_t|X_t, z_{1:t-1})$ two main groups of solutions can be found: the Kalman filtering methods and the sampling methods.

Optimal solutions: Kalman filters

The Bayesian tracking problem has an optimal solution in the restrictive cases where three following hypotheses hold:

1. The state and observation noise processes η_t and ν_t are zero-mean Gaussian distribution with known covariance matrixes Σ_t and Λ_t.

2. The state evolution function $f_t(X_{t-1}, \eta_{t-1})$ is a known linear function of X_{t-1} and η_{t-1}:

$$X_t = F_t X_{t-1} + \eta_{t-1}. \tag{2.17}$$

3. The observation function $h_t(X_t, \nu_t)$ is a known linear function of X_t and ν_t:

$$z_t = H_t X_t + \nu_t. \tag{2.18}$$

When these hypotheses hold, the Kalman filter is the explicit optimal solution of the tracking problem [68]. Given the three previous assumptions and assuming an initial Gaussian distribution for $p(X_0)$, the pdf of the state at any given time, given measurements up to time t is a Gaussian distribution

$$p(X_t|z_{1:t}) = \mathcal{N}(X_t, m_{t,t}, P_{t,t}) \tag{2.19}$$

where $m_{t,t}$ and $P_{t,t}$ denote the mean and covariance of the this optimal distribution. The update and prediction steps in Equations 2.12 and 2.16 can be rewritten as:

$$\begin{aligned} &p(X_t|z_{1:t-1}) = \mathcal{N}(X_t, m_{t,t-1}, P_{t,t-1}) \\ &p(X_t|z_{1:t}) = \mathcal{N}(X_t, m_{t,t}, P_{t,t}) \end{aligned} \tag{2.20}$$

where

$$\begin{aligned} &m_{t,t-1} = F_t m_{t-1,t-1} \\ &P_{t,t-1} = \Sigma_{t-1} + F_t P_{t-1,t-1} F_t^T \\ &m_{t,t} = m_{t,t-1} + K_t(z_t - H_t m_{t,t-1}) \\ &P_{t,t} = P_{t,t-1} + K_t H_t P_{t,t-1} \end{aligned} \tag{2.21}$$

with K_t and S_t defined as

$$\begin{aligned} &K_t = P_{t,t-1} H_t^T S_t^{-1} \\ &S_t = H_t P_{t,t-1} H_t^T + \Lambda_t \end{aligned} \tag{2.22}$$

K_t is the Kalman, gain and S_t is the covariance matrix of the innovation term $z_t - H_t m_{t,t-1}$.

In other cases, where the three assumptions to obtain an optimal solution are not valid, approximate solutions have to be found. When the dynamic function f_t and the observation function h_t are well represented by their first order linear approximations, and if the filtering pdf can be approximated by a Gaussian distribution, a sub-optimal solution is given by the extended Kalman filter (EKF) [69]. The drawback of the EKF is related to the linearization of the functions f_t and h_t. In general, the state evolution function, f_t, admits good linear approximation. The problems occur, in general, with the observation function, h_t, which usually very complex. If the first order linearization do not approximate well the non-linearities in the observation function, the EKF will introduce large errors in the mean and covariance matrices estimates, and sometimes will lead to the filter divergence. If the function h_t is known and well represented by its third order Taylor series expansion, the unscented Kalman filter (UKF) proposed in [70, 71] is a good sub-optimal solution to the tracking problem. The UKF is based on the unscented transformation which is a method to calculate the statistics of random variables undergoing non-linear transformations using a minimal set of carefully chosen sample points [72].

However in real life tracking, the analytic form of the function h_t is in general unknown, and when it is known, does not admit good linear representation. Also due to occlusions and ambiguities, pdf are not Gaussian. They are multi-modal or heavily skewed, hence the conditions to obtain optimal or suboptimal solution do not hold in general. Thus, other solutions such as the sequential Monte carlo methods are needed.

Approximate solutions: sequential Monte Carlo (SMC) methods

SMC methods, also known as particle filters (PF), are methods to approximate the pdf when the conditions to obtain optimal or sub-optimal conditions do not hold [69, 73, 74, 75]. They are techniques to implement a recursive Bayesian filter by Monte Carlo simulation. The key idea is to represent the target pdf $p(X_t|z_{1:t})$ by a set of random samples called particles with associated weights, as illustrated by Figure 2.11. Let $\{X_t^{(n)}, w_t^{(n)}\}_{n=1}^{N_s}$ denote a set of weighted samples drawn from the pdf $p(X_t|z_{1:t})$, with the weights $w_t^{(n)}$ normalized to sum to 1. An approximation of the true pdf is given by:

$$p(X_t|z_{1:t}) \approx \sum_{n=1}^{N_s} w_t^{(n)} \delta_{X_t^{(n)}}(X_t) \tag{2.23}$$

where δ is defined as

$$\delta_{X_t^{(n)}}(X_t) = \begin{cases} 1 \text{ if } X_t^{(n)} = X_t \\ 0 \text{ otherwise} \end{cases} . \tag{2.24}$$

When the number of samples becomes very large, the PF solution approximates the true pdf [74]. The goal of the PF algorithm is the recursive propagation of the samples and estimation of the associated weights, as each measurement is received sequentially. To this end, a recursive equation of the posterior is used:

$$p(X_t|z_{1:t}) = \frac{p(z_t|X_t, z_{1:t-1}) \int_{X_{t-1}} p(X_t|X_{t-1}, z_{1:t-1}) p(X_{t-1}|z_{1:t-1}) dX_{t-1}}{p(z_t|z_{1:t-1})}. \tag{2.25}$$

Assuming that the pdf $p(X_{t-1}|z_{1:t-1})$ can be approximated with samples according to Equation 2.23, the current pdf can be approximated as:

$$p(X_t|z_{1:t}) \approx \frac{p(z_t|X_t, z_{1:t-1}) \sum_n w_{t-1}^{(n)} p(X_t|X_{t-1}^{(n)}, z_{1:t-1})}{p(z_t|z_{1:t-1})}. \tag{2.26}$$

Let us denote by $q(X_t|X_{t-1}, z_{1:t})$ a distribution, called importance or proposal density, from which new hypotheses can be easily sampled. The pdf in Equation 2.26 can be approximated, up to the proportionality constant $p(z_t|z_{1:t-1})$, as:

$$p(X_t|z_{1:t}) \approx \sum_n w_{t-1}^{(n)} \frac{p(z_t|X_t, z_{1:t-1}) p(X_t|X_{t-1}^{(n)}, z_{1:t-1})}{q(X_t|X_{t-1}^{(n)}, z_{1:t})} q(X_t|X_{t-1}^{(n)}, z_{1:t}) \tag{2.27}$$

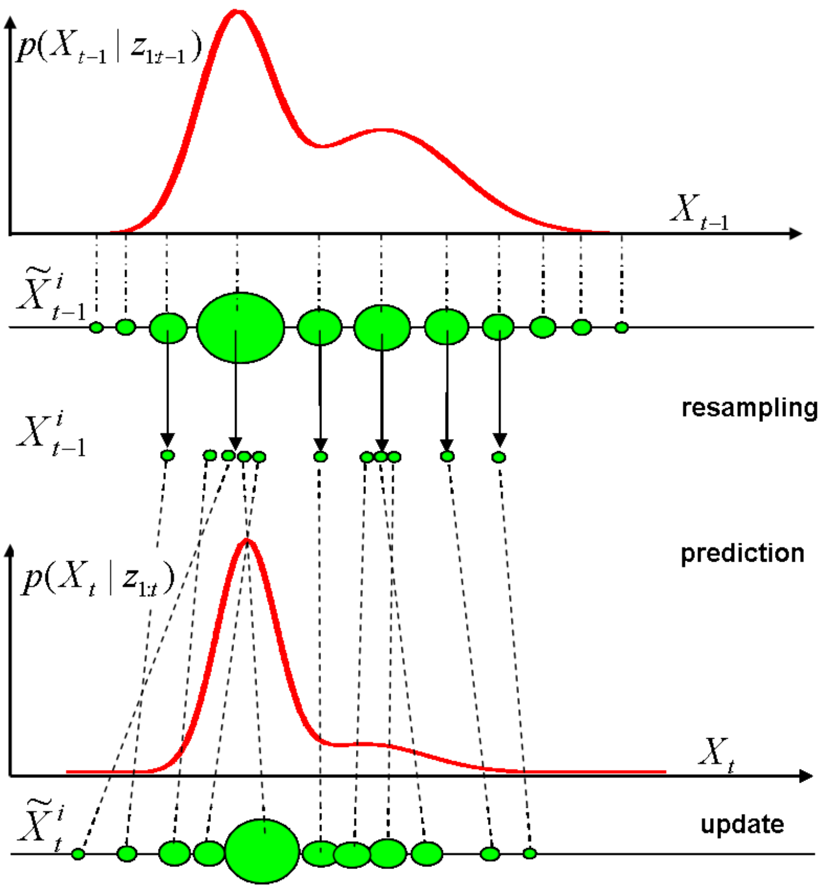


Figure 2.11: SMC steps: approximation of a continuous probability density function with discrete weighted samples. Propagation of the samples and weights update.

Thus, using factor sampling, a discrete representation of the pdf can be obtained in two steps:

1. First, predict new hypotheses using the proposal distribution:

$$X_t^{(n)} \sim q(X_t | X_{t-1}^{(n)}, z_{1:t}) \tag{2.28}$$

2. Then, update the sample weights according to:

$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(z_t | X_t^{(n)}, z_{1:t-1}) p(X_t^{(n)} | X_{t-1}^{(n)}, z_{1:t-1})}{q(X_t^{(n)} | X_{t-1}^{(n)}, z_{1:t})} \tag{2.29}$$

Depending on the assumptions on the above distributions, various algorithms can be obtained. The simplest one is the sequential importance resampling (SIR) filter, also known as the conditional density propagation (CONDENSATION) algorithm [13] in the computer vision literature.

Sequential importance resampling (SIR) algorithm:
To obtain the standard PF algorithm, the three following hypotheses are made.

1. First, the state dynamic is a first order Markov model:

$$p(X_t | z_{1:t-1}, X_{t-1}) = p(X_t | X_{t-1}) \tag{2.30}$$

2. Secondly, given the sequence of states, the observations are independent. Hence we have

$$p(z_t | z_{1:t-1}, X_{1:t}) = p(z_t | X_t). \tag{2.31}$$

3. Thirdly, the state dynamic, $p(X_t | X_{t-1})$, is used as importance function.

The two first hypotheses corresponds to the standard tracking graphical model displayed in Figure 2.10. Ultimately, from the three previous hypotheses, a simple update equation for the weights is obtained as

$$w_t^{(n)} \propto w_{t-1}^{(n)} p(z_t | X_t^{(n)}). \tag{2.32}$$

This algorithm is called sequential importance sampling (SIS) algorithm [69]. A common problem with the SIS algorithm is the degeneracy phenomenon. After a few iterations, most of the particles have very low weights. It has been shown that the variance of the weights can only increase over time [74]. Thus, it is impossible to avoid the degeneracy phenomenon. A large computational effort is devoted to update particles whose contribution to the approximation of the pdf are very negligible. A method to reduce the effect of the degeneracy is to resample with replacement the particles with a probability proportional to their importance weights, in order to keep only those with high weights [76]. The SIS algorithm including a resampling step is called sequential importance resampling (SIR) algorithm. This algorithm is also known as bootstrap filter [73], (CONDENSATION algorithm [13]. Figure 2.12 summarizes the main steps of the SIR algorithm.

A limitation of the SIR algorithm is related to the choice of the state dynamic as proposal distribution. This choice implies that new hypotheses are generated without taking into account the new observation

1. **Initialization Step:**
   for $n = 1, ..., N_s: \ X_0^{(n)} \sim p(X_0), \ w_0^{(n)} = \frac{1}{N_s}, \ t = 1.$
2. **Importance Sampling Step:**
   - $\tilde{X}_t^{(n)} \sim p(X_t|X_{t-1}^{(n)}), \ \tilde{w}_t^{(n)} \propto p(z_t|\tilde{X_t^{(n)}}), \ n = 1, ..., N_s$
   - $\tilde{w} = \sum_n \tilde{w}_t^{(n)}, \ \tilde{w}_t^{(n)} = \frac{\tilde{w}_t^{(n)}}{\tilde{w}}, \ n = 1, ..., N_s$
3. **Resampling Step:**
   Resample with replacement the particles according to their weights. $\{\tilde{X}_t^{(n)}, \tilde{w}_t^{(n)}\} \longrightarrow \{X_t^{(n)}, w_t^{(n)} = \frac{1}{N_s}\}$
4. **Time Increment Step:**
   set $t = t + 1$, the GO TO Step 2

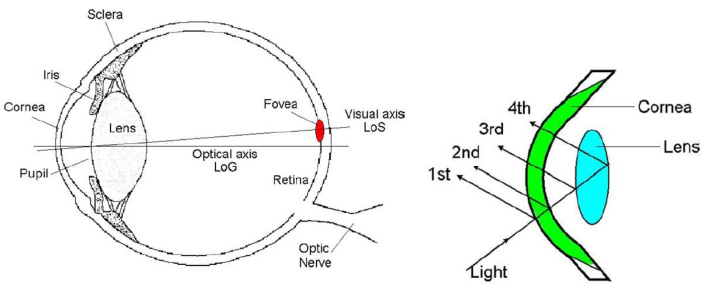
Figure 2.12: Sequential Importance Resampling (SIR) Algorithm.

z_t which is available. Several approaches have been proposed to address this issue. For instance, when available, auxiliary information generated from color [77], motion [15, 77], or audio in the case of speaker tracking [78], can be used to draw samples from. The proposal distribution is then expressed as a mixture between the dynamic and state detectors based on color, motion, or audio observations. In our work, we follow this approach of sampling from the dynamic and a detector. An advantage of this approach is that it allows for automatic re-initialization after tracking failures. But it has as drawback that, the additional samples, proposed by the detector, are not related anymore to the samples at the previous time. Thus, this approach requires, for each new sample, an expensive computation of a transition prior involving all the previous samples [77, 78, 79]. Another auxiliary particle filter proposed in [80] avoid the problem related to sampling from the dynamic. Their idea is to use the likelihood of a first set of particles at time $t-1$ to resample the seed samples at time t, and apply standard prediction and evaluation steps on these seed samples. The feedback from the new data acts by increasing or decreasing the number of descendant of a sample depending on its predictive likelihood.

Finally, lets mention that other SMC methods such as Rao-Blackwellized PF and Markov chains Monte Carlo (MCMC) methods will be described in more details in Section 4.

2.3 Visual focus of attention (VFOA)

The VFOA of a person can be defined as the person or the object a person is focusing his visual attention on. Thus, the VFOA of a person can be inferred from the direction in which his eyes are pointing, which is called the eye gaze. For human being, vision is the main perceptual sense. Thus studying the gaze and VFOA of people is important to understand human behaviors. There are two domains where the role of gaze has been widely and thoroughly studied: the first one is the visual search. Domain in which the goal is to understand how people analyze their visual environment. The second domain, more related to our research is the gaze in social interaction, where the goal is to study the pattern and the role of gaze in human communication.



(a) Eye structure. (b) Purkinge images.

Figure 2.13: Eye structure and Purkinge images (from [81]).

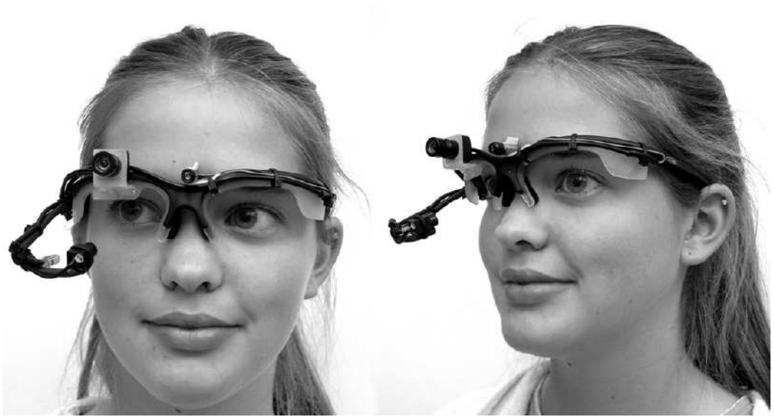


Figure 2.14: Light weight gaze tracking system (from [82]).

In the following we will first review some sensor based gazed tracking methods and then discuss related work about gaze in visual search and gaze in social interactions.

2.3.1 Sensor based gaze tracking systems

Sensor based techniques can be grouped into three families: techniques based on reflected light, techniques based on electric skin potential measurements, and techniques based on contact lenses.

The reflected light eye gaze estimation techniques rely on the amount and direction of the infrared light reflected by specific parts of the eye such as the limbus (boundary between the white sclera and the black iris) displayed in in Figure 2.13(a), the pupil, the corneal. The reflections from the different components of the eyes create the Purkinje images, shown in Figure 2.13(b), that can be used to estimate the eye gaze [83, 84, 82] .

The electric skin potential technique is based on the fact that there exist an electrostatic field which is rotating with the eye ball. By recording the difference in the skin potential around the eye, the position of the eye in the eye globe can be inferred [85].

Gaze direction can also be estimated by asking a subject to wear special contact lenses [86]. There are two kinds of contact lenses. The first type of lenses has mirror-like surface reflecting an infrared light beam shined on the eye direction and allowing the eye gaze estimation. The second type of lenses are inducted with matter sensitive to magnetic field. When the head of the subject is placed inside an electric field, the distortion of the magnetic field due to the lens position in the eye allows to estimate the gaze direction. Estimating gaze with electro-magnetic sensor is sensitive in situations where there exist objects modifying the electro-magnetic field such as computers, cell-phones or even iron tables.

The main advantage of sensor based techniques is their precision in the gaze direction estimation. The gaze direction can be estimated at precisions sometimes less than 1 degree [81]. The major drawbacks of the sensor based gaze tracking systems is their invasiveness. The subject has either to wear sensors such as electrodes, lenses or helmet mounted with mirrors. Also, they can be quite restrictive because they need to be calibrated with respect to the head location. Although in general sensor based techniques require the head to be steady, systems such as the light weight gaze tracking systems, in Figure 2.14, allows more head motion.

2.3.2 Gaze in visual search

Gaze scan paths during visual search have been studied for long to understand the visual perception and recognition of humans [87]. Many studies have shown that, in non-oriented task situations, that eye fixation locations are, in general, points of high local contrast [88]. However, in task-oriented situations, the eyes fixations were shown not to always correspond to the visually most salient points, but to follow specific spatio-temporal requirements in order to full-fill the task. Investigations in this sense have been made in visio-motor tasks such as driving [89]. In [89], the scan paths of experienced and novice drivers were studied in order to identify differences that can be associated with skill acquisition and sensitivity to road type. Similar study conducted in daily life activities such has making tea or sandwich [90] showed that the fixation points are tightly linked in time to the to the evolution of the task. Scan paths are important because they give indirect information about a person cognitive process. Other works, such as [91], showed that eye movements are so tightly connected to the cognitive processes that, in well defined tasks, the eyes movements can be used to observe under natural conditions the mental processes that underlie spoken language comprehension.

For advertising also, scan paths are important to study. Information about the path of the gaze can be used to define locations to display information an advertiser wants consumers not to miss. For instance, the study of consumers scan paths showed that for a brand, the pictorial of the brand was attracting the consumers'gaze more than a text about the brand [92]. Furthermore, in [93] it was also shown that the scan paths can be used to determine highly motivated consumers and the brands they are interested in.

One of the main problem of studying gaze in natural conditions is that the current gaze tracking systems, although portable such as the light weight gaze tracker in Figure 2.14, are still constraining. It is difficult to imagine in natural meetings or in an outdoor marketing study people wearing sensor based gaze tracking systems. Also, gaze paths are not only the result of eye movements but also of head movements. In [94], the authors have shown that in reading situations, the head was contributing to 40.3% of the gaze horizontal gaze amplitude and 28.4% of the vertical gaze amplitude. Thus, gaze has to be studied together with head movements.

2.3.3 Gaze in social interactions

A major domain where gaze plays an important role is human interaction. Interacting with other people is an essential component of social activity. The way in which these interactions occur in groups is the topic of intense study in social psychology [95]. The gaze is an important cues used in human interaction.

Gaze fulfill functions such as establishing relationship (through mutual gaze), regulating the course of interaction, expressing intimacy [96], and exercising social control [1]. Humans tend to look at thing that are of immediate interest for them. Information from gaze, head and body posture are used to determine where the other people are directing their attention. Thus gaze helps to predict the other person mental states, and people use their gaze to give to the other insight about what they are thinking. For instance, in [97], it was shown that in 4 people conversations, when a speaker gaze another person in the eye, at 80% this person, he is gazing, is the target of his speech. Furthermore, when a listener is gazing at a person in the eye in a meeting, at 77%, this person was the speaker. Thus people use the other's gaze to determine when they were addressed or expected to speak. A speaker's gaze often correlates with his addressees gaze, especially at a sentence end where the gaze can be interpreted as a request of back-channel [98]. Studies have shown that people take more turns when they experienced more gaze [99]. For a listener, monitoring his own gaze in concordance with the speaker's gaze is a way to find appropriate time windows for speaker turn requests [100, 101].

Due to all the information conveyed by the gaze in social interaction, its study is important to understand human interaction contents. Sensor based gaze tracking techniques can be used to study the gaze during interaction. However, due to the invasiveness of the usage of sensors for gaze tracking, computer vision techniques are, in some cases, better suited for the gaze estimation. Although less precise than gaze tracking with sensors, computer vision techniques will make possible the gaze study in all the available pre-recorded video. In thesis, our interest will be to study gaze from head pose using computer vision techniques.

2.4 Conclusions

In this chapter we have discussed the state of the art related to the main topics of this thesis. Various head models were presented, followed by tracking methodologies and gaze studies. In this thesis we will present our head pose tracking methodologies. But, as head pose tracking requires head pose tracking evaluation, the following Section present our head pose video database built for head pose tracking evaluation.

Chapter 3

Head pose video database

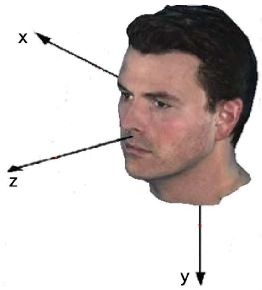
As already discussed in the introductory chapter, tracking the head of people and estimating their pose is important in many computer vision applications. This has generated a large amount of investigations in the related research fields especially since good automatic analysis of head behaviors rely on precise head tracking and pose estimation. The performance of the tracking methods have to be rigorously evaluated and this requires the availability of significant public head pose databases.

In the vision community, still head pose image databases exist such as the FERET database [46], the PIE database [47] and the Prima-Pointing database [48]. Still head pose image databases are very useful to build head pose models and evaluate head pose detection or recognition algorithms. However, the evaluation of head pose tracking algorithms requires realistic video sequences with people having their head poses continuously annotated. As head pose annotation in video sequences is a non-trivial task, most of the time, head pose tracking algorithms are evaluated qualitatively on video sequences without head pose annotations. Some researchers used head pose video databases to evaluate their algorithms [44, 102], but their database is not publicly available. Recently, for the CLEAR Evaluation workshop a database for head pose tracking was built and made publicly available [103]. The limitation of the CLEAR data is that the head poses in the database were not continuous: 8 ranges of head directions (North, North-East, East,..., where the cardinal directions correspond to the wall of a room) were defined and annotated by hand. Thus, in short, comparing head pose tracking performances on continuous video sequences is difficult, because of a lack of evaluation data. As researchers have been working on head pose tracking since a long time, a publicly available database to evaluate head pose tracking methods and compare performances is required.

This chapter describes our work to build and provide such a video database, featuring people in real situations with their head pose continuously annotated through time. To build a head pose database, a protocol for head pose annotations has to be defined. In our case, head poses were annotated using a magnetic 3D location and orientation tracker called the flock of bird [104]. The environments of our recordings were a meeting room and an office with their light sources. These environments summarize well indoor environments for head pose tracking. The recording in the meeting room involved 16 persons, 2 persons per meeting, and each meeting lasted approximatively 10 minutes. The office recording involved also 15 persons, with 1 person per recording, and recording lasted approximatively 8 minutes.

The remaining of this Chapter is organized as follows. Section 3.1 describes possible head pose representation. Section 3.2 describes our database recording set up. Section 3.3 describes the database

(a) Reference head configuration $\mathcal{R}_0$: frontal head pose



(b) Head attached basis $\mathcal{B}_{head}$.

Figure 3.1: References for head pose representation. Figure 3.1(a) displays the frontal pose, and Figure 3.1(b) gives the head attached basis.

content. Finally, Section 3.4 gives conclusions.

3.1 Head pose representation

A head pose representation is a parametrization of the head configuration with respect to a given reference $\mathcal{B}_0$ and a reference head configuration $\mathcal{R}_0$. Figure 3.1(a) shows our reference head configuration which is taken to be the head in a frontal pose. In the following we always assume that a basis $\mathcal{B}_{head}$ is rigidly attached to the head, as can be seen in Figure 3.1(b). A basis $\mathcal{B}_{cam}$ is also attached to the camera. By definition, the head configuration can be represented by the 3×3 rotation matrix R_θ that allows to transform the axis of the basis attached to the head associated with $\mathcal{B}_0$ into the axis rigidly attached to the head at the current configuration.

In our database, we defined the head poses with respect to the camera viewing direction, not with respect to a fixed and absolute reference basis. Thus, the annotated head poses can be directly compared with the output of head pose trackers, which primarily output this kind of estimations. In practice, this means that a person looking at the camera, whether on the left or on the right of the image, will be always annotated with a frontal pose. despite the fact that their head poses are oriented differently with respect to an absolute 3D reference in the room.

As an alternative to the matrix representation of head rotation, it is well known that rotation matrices in 3D can be represented as a succession of 3 rotations around the 3 axis of a reference basis $\mathcal{B}$: a rotation around the y-axis R_α, a rotation around the x-axis R_β and a rotation around the z-axis R_γ. The 3 angles of such a decomposition, $\theta = (\alpha, \beta, \gamma)$, are called Euler angles [105]. Among the multiple possible Euler angle parameterizations, two are commonly used. The first one uses as reference basis for the rotation axis, the basis attached to the head, $\mathcal{B}_{head}$. This representation was used to build the FERET database [46] and the Prima-Pointing database [48]. A second parametrization uses the camera attached basis, $\mathcal{B}_{cam}$, as reference frame for the rotation axis. This representation was used to build the PIE database [47]. In the following, we describe these two head pose representations and give ways to pass from one representation to another.

Figure 3.2: Poses in the Pointing representation (from Prima-Pointing database [48]).

3.1.1 The pointing representation

This representation has been used to build the Prima-Pointing database. The Prima-Pointing database will be described in more detail in Section 4. Figure 3.2 shows sample head poses of a person in this database. In the pointing representation, a head pose is defined by three Euler angles $\theta = (\alpha, \beta, \gamma)$ representing three consecutive rotations to transform the head from the reference configuration $\mathcal{R}_0$ to a current configuration. The rotations are done with respect to the axes of the basis $\mathcal{B}_{head}$ rigidly attached to the head. The pan rotation α is a left-right rotation, the tilt rotation β is an up-down rotation, and the roll rotation γ is a head-on-shoulder roll. This representation can be thought as if the camera was fixed and the head rotating. The pointing representation is interesting because it is very intuitive and gives interpretable values. The poses obtained from this representation correspond to the natural ways people perceive head rotations.

3.1.2 The PIE representation

The setup used to build the PIE database was the following: 9 of 13 cameras were positioned at a roughly head height in an arc from approximately a full left profile to a full right profile. Each neighboring pairs of these 9 cameras were therefore approximately 22.5 degrees apart. Of the remaining 4 cameras, 2 were placed above and below the central (frontal) camera, and 2 were placed in the corners of the room. Images of the views of the 13 camera are displayed in Figure 3.3. In the PIE representation, the pose of a person is also defined relatively to a reference head configuration $\mathcal{R}_0$. A head pose is defined by three Euler angles $\theta = (\alpha, \beta, \gamma)$. These angles define three consecutive rotations that, when applied to the head, transform the reference head configuration into the current configuration. But, in the PIE representation, the rotations are with respect to the static axes of the camera reference $\mathcal{B}_{cam}$, pointing towards the head direction.

Perceptually, the PIE representation can be seen as if the head was static and the camera rotating around it. The PIE representation has a computational advantage. In this representation, head roll variations for given pan and tilt angles correspond to an image in plane rotation. More precisely, given the image appearance of the head pose $\theta = (\alpha, \beta, 0)$, the image appearance of any pose $\theta = (\alpha, \beta, \gamma)$ is obtained by applying an in-plane rotation of angle γ to the image appearance of the pose $\theta = (\alpha, \beta, 0)$. This property, which does not hold for the pointing representation, is very useful when building head pose appearance models. The variation in head roll do not need to be modeled in the PIE representation. The drawback of the PIE representation is that the significance of the Euler angles are not anymore in-

Figure 3.3: Poses in the PIE representation (from CMUPIE database [47]).

terpretable. For instance, the tilt at a frontal pose are similar than in the pointing representation but, the tilt at a profile pose in the pointing representation corresponds to a roll in the PIE representation.

3.1.3 From one representation to another

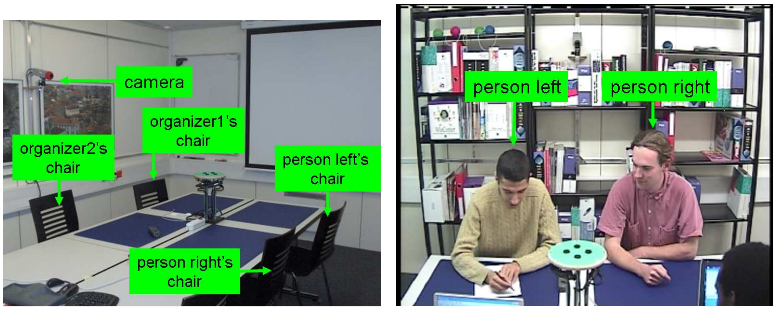
Each one of the two representations have advantages and drawbacks. Depending on the problem, we may be interested in using one of the representations. Thus, being able to convert the head pose in a given representation into the other representation is useful.

If we denote by θ a head pose in the PIE representation with its corresponding rotation matrix R_θ. Also, let denote by θ^* the same pose in the pointing representation, with its corresponding rotation matrix, R_{θ^*}. Since the two head poses representation, θ and θ^* represent the same head rotation, they are represented by the same rotation matrix. Thus, to pass from one representation to the other, one has to find θ^* (resp. θ) such that $R_{\theta^*} = R_\theta$.

3.2 Database building

3.2.1 Camera calibration

The database was recorded in two environments, a meeting room and an office. Figures 3.6 and 3.4 show the recording setup, camera and people location. For each environment, one camera was used for the recordings. The camera in the office was mounted in the horizontal plane at head height and was recording at 12.5 image frames per second. The meeting room camera is located at around 60cm upper the head height, pointing down with an optical axis making approximately a 15 degree angle with respect to the horizontal plane, and was recording at 25 image frames per second. The cameras were fixed, and head poses were defined with respect to the reference head configuration as shown in Figure 3.1(a). The cameras were calibrated using the methodology described in [106]. The principle of the camera calibration procedure was to show the camera a checkerboard viewed at many orientations as illustrated in Figure 3.5.



(a) Meeting room setup.

(b) Example image from a meeting room recording.

Figure 3.4: Database recording set-up: camera and people locations in the meeting room recordings.

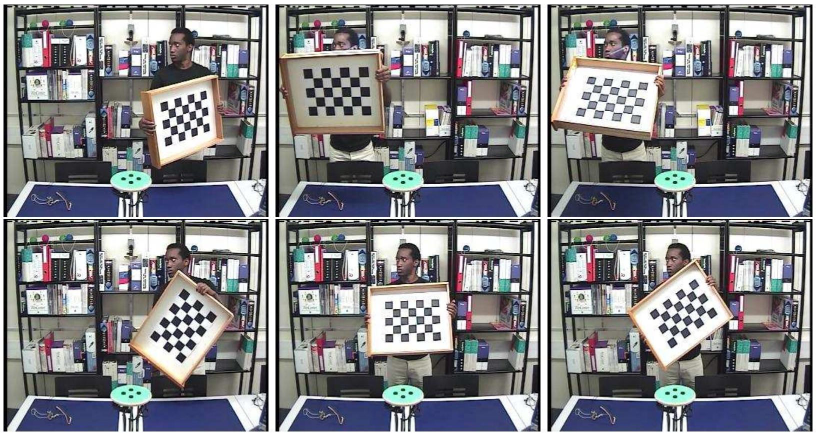


Figure 3.5: Camera calibration: part of the checkerboard views used to calibrate the camera in the meeting room.

The structures of the checkerboard, number and size of squares, were used to find the correspondence between projected images and the 3D geometry of the room. The outputs of the camera calibration are the extrinsic and the intrinsic parameters of the camera. Given a point P^{ex} in an external 3D reference $\mathcal{B}_{ex}$, the external camera parameters (t^{ex}, R^{ex}) are a translation and a rotation matrix defining the change basis to pass from the external reference $\mathcal{B}_{ex}$ to the camera reference. The relation between a point, P^{cam}, in the camera reference and its representation, P^{ex} in the external reference is given by:

$$P^{cam} = t^{ex} + R^{ex} P^{ex} \tag{3.1}$$

The intrinsic camera parameters define how the point P^{cam} in the camera basis projects in the image plane. The intrinsic camera parameters are defined by four elements, the focal length $fc = (fc^1, fc^2)^T$, the principal point $cc = (cc^1, cc^2)^T$, the skewness distortion ac, and the radial distortion $kc = (kc^1, kc^2)^T$.

(a) Office setup

(b) Example image from an office recording.

Figure 3.6: Database recording set-up: camera and people locations in the office recordings.

In the general image projection framework, kc is a four dimensional vector defined by the radial distortions and the tangential distortions [106]. The tangential distortions are null in our case, thus they are removed from the Equations. The point P^{cam} projects on the image plane according to the intrinsic parameters as P^{im}, with the corresponding coordinates:

$$\begin{aligned} P^{im,1} &= fc^1(P^{dist,1} + acP^{dist,2}) + cc^1 \\ P^{im,2} &= fc^2 P^{dist,2} + cc^2 \end{aligned} \tag{3.2}$$

If we denote the coordinates of the point in the camera reference as

$$P^{cam} = (P^{cam,1}, P^{cam,2}, P^{cam,3})^T, \tag{3.3}$$

P^{dist} is then defined as:

$$P^{dist} = (1 + kc(1)r^2 + kc(2)r^4)\left(\frac{P^{cam,1}}{P^{cam,3}}, \frac{P^{cam,2}}{P^{cam,3}}\right)^T \tag{3.4}$$

with $r^2 = \left(\frac{P^{cam,1}}{P^{cam,3}}\right)^2 + \left(\frac{P^{cam,2}}{P^{cam,3}}\right)^2$ The intrinsic camera parameter values are given in Table 3.1. The most important parameter for reconstruction quality are the distortion parameter kc. Its components have to be small with respect to 1 for a good reconstruction. For the meeting room camera the components of the distortion parameter are small. However, for the office camera it is worth noticing that the distortions are not negligible when a point is far from the focal point (r^2 is high).

3.2.2 Head pose annotation with the flock of bird sensor

Flock of Bird Measurements: We used the pointing representation to build our head pose video database. For the head pose annotation we used a device called flock of bird (FOB) [104]. The FOB is a 3D location and orientation magnetic field tracker with two main components. A reference basis unit, rigidly attached to the desk, and a bird rigidly attached to the head of the person whose head pose

camera	meeting room	office
fc	(425 462)	(416,453)
cc	(174,130)	(220,127)
ac	0	0
kc	(-0.27,-0.07)	(-0.36,0.49)

Table 3.1: Intrinsic camera parameter values.

has to be annotated. The FOB outputs locations and orientations with respect to its reference basis.

FOB-camera calibration: Calibrating the camera to the FOB correspond to finding a translation vector and a rotation matrix (t^{fob}, R^{fob}) such that any point in the Camera basis and it representation in the FOB are related by the formula:

$$P^{cam} = t^{fob} + R^{fob} P^{fob}. \tag{3.5}$$

The rotation matrix and the translation vector (t^{fob}, R^{fob}) can be approximated numerically by solving an optimization problem. Given a set of points $\{P_i^{cam}, \, i = 1, ..., N\}$ in the camera basis with their known corresponding representations in the FOB basis $\{P_i^{fob}, \, i = 1, ..., N\}$, the problem is to estimate $(\hat{t}^{fob}, \hat{R}^{fob})$ such that:

$$(\hat{t}^{fob}, \hat{R}^{fob}) = \arg \min_{(t^{fob}, R^{fob})} \sum_{i=1}^{N} ||P_i^{cam} - (t^{fob} + R^{fob} P_i^{fob})||_2 \tag{3.6}$$

where $||.||_2$ is the Euclidean distance. This problem can be solved using gradient descent.

Temporal alignments: In our setup, the camera and the FOB recordings were started manually. There was a time delay between the recording starting time of the two devices. The FOB outputs and the video frames have to be aligned. The alignment can be done by finding an easy-to-identify event in the video sequence and in the FOB data. We defined the alignment events to be quick head shake. This gesture corresponds to an oscillation of the head pan while the head tilt and roll are steady in the FOB data. The video frames corresponding to the peaks of this oscillation are also easy to find in the video sequences.

3.3 Database content

The database is constituted by recordings in two environments, a meeting room and an office. In the following we describe the recording contents in these two environments.

3.3.1 Meeting room recordings

In the meeting room, 8 meetings were recorded. In each recording, 4 persons were having a meeting and among them, two had their head pose continuously annotated. The durations of the meetings are given in Table 3.2. These meetings were recorded according to a simple scenario. The people had to look at the

Meeting	1	2	3	4	5	6	7	8
duration	7.6	7.5	7.3	11	10	10.9	14.3	12.1

Table 3.2: Meeting room recordings durations (in minutes).

camera in a frontal head pose to define the head reference configuration, perform the alignment gesture, write their name on a sheet of paper on the table and discuss statements displayed on the projection screen. The scenario gives full freedom to the participants about their head motion, pose and gestures. People were acting naturally as in real meeting situations. The meeting lengthes vary between 7.6 to 14 minutes, thus studying the visual focus of attention in these recording will be interesting. The recordings are long enough to exhibit a wide range of gazing behaviors. An image of each of the 8 meetings is provided in Figure 3.7.

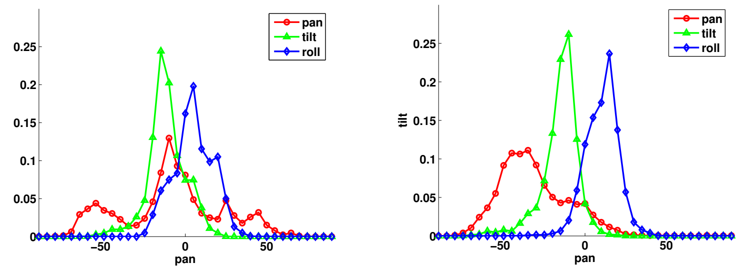
Figure 3.8 gives the distribution of head pan, tilt, and roll angles over the meeting data set. The pan values are ranging within -90 and 50 with several modes. The tilt values are ranging from -50 to 20, and the roll values from -20 to 40 degrees. Figure 3.9 displays the head pan versus head tilt scatter plots for the two persons in the first meeting recording. It can be seen in these Figures 3.8 and 3.9, that the two persons, mostly the person sitting to the right, had negative pan values. Negative pan values corresponds mainly to looking at the projection screen which was an important visual focus of attention for the persons according to our scenario. It has also to be noticed that the way the camera was mounted, 60 cm upper head location and 15 degrees with respect to horizontal, was inducing a shift in the tilt values. When a person is looking straight in front of him in the horizontal plane, his head tilt is -15 degrees while it would be 0 degree if the camera would have been mounted at head height and pointing horizontally.

3.3.2 Office recordings

The office recordings involved 15 persons. The length of each recording is given in Table 3.3. In each recording, a person was sitting in front of a computer and acting in the framework of a simple scenario. First look at the camera in a frontal head pose and perform an alignment gesture. Then, look at fixed points of the room and finally, interact with the experimenter. The office recording set up was close to a human computer interaction (HCI) set up. The head image sizes were quite high, varying approximately between 100×100 and 180×180 pixels.

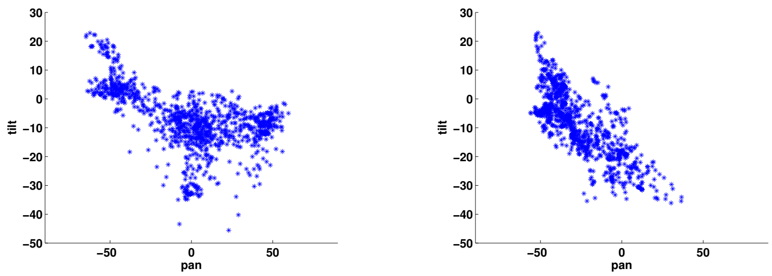
For each recording, the flock recordings, after alignment and transformations using the FOB-camera calibration procedure, give the head pose annotations. Figure 3.10(a) shows the distribution of the head pan, tilt, and roll values for the whole office recordings. Pan value are ranging from -50 to 150 degrees in a quite flat distribution. The tilt values are ranging from -60 to 20 degrees but most of the values are within -15 and 15. Roll values are ranging from -20 to 20 degrees with the values mostly concentrated between -5 and 10 degrees. Figure 3.10(b) displays the scatter plot of the head pan versus head tilt of the subject in the first office recording. In this plot, we can notice that the pan are mostly positive. The reason is that the experimenter (person in the background in Figure 3.6(b)) was sitting on the side of the positive pan (right side) of the annotated person.

Figure 3.7: The 16 annotated people from the 8 meetings.



(a) Distribution for person left. (b) Distribution for person right.

Figure 3.8: Distribution of head pan, tilt, and roll in the meeting recordings, expressed in the pointing representation.



(a) Plots for person left in the first meeting recording. (b) Plots for person right in the first meeting recording.

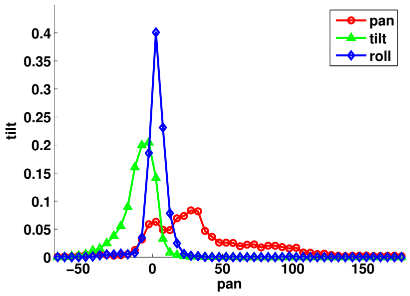
Figure 3.9: Head pan-tilt scatter plot for person left and right in the first meeting room recording.

3.4 Conclusion

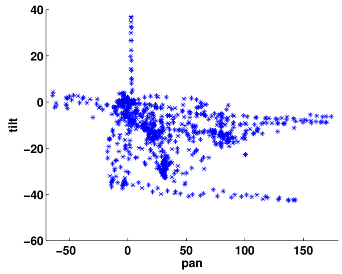
In this chapter, we have described our head pose video database. This database was built to achieve three goals. First, build for ourselves a head pose video database with people having their head orientation continuously annotated to evaluate our own head pose tracking algorithms. Second, distribute it to the scientific community to be used as evaluation database. Information about the database, and the procedure to acquire the database are available at *http://www.idiap.ch/HeadPoseDatabase/*. This could be very useful for the head pose tracking community in which, in general, people evaluate their head pose tracking algorithms either qualitatively or on private data making performance comparison impossible. In the following chapter, we will present our head pose tracking algorithms and use the meeting room recordings to evaluate our algorithms. For the evaluations, we will follow a precise protocol to allow comparisons with other approaches.

Office	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
duration	4.6	4	5.5	4.3	5.6	7	6.2	5.5	5.6	5.6	7.2	7.1	5.4	6.3	6.7

Table 3.3: Office recording duration in minutes.



(a) Pose angle distributions for the subject in the office recordings.



(b) Head pan-tilt scatter plot for the subject in the first office recording.

Figure 3.10: Head pose distribution in the office recordings.

Chapter 4

Head pose tracking

Head pose tracking has been widely investigated by the computer vision research community. Initially, most of the proposed methods, refer to as model based methods, were based on facial features tracking and 3D head pose geometric reconstruction from the location of the tracked facial features. But, tracking facial features usually requires high resolution images. Also, because of ambiguities and occlusions, tracking facial features is likely to fail in long term tracking. Thus, other methods called appearance based methods were proposed. These methods are based on global head appearance models learned from training data using machine learning techniques as discussed in Section 2.1. The head pose tracking methods can also be classified into two categories according to wether the tracking and the pose estimation are done jointly or not. In the first category, the head is tracked and its location found. Then, the image around the head location is processed to estimate the head pose. This methodology, by neglecting the relationship between the head spatial configuration (location, size) and the pose, leads to a reduction of the computational cost of the algorithms by reducing the dimensionality of the search space. This aspect of the problem is important when trying to design real time systems. However, head pose estimation has been shown to be very sensitive to the precision of the head localization. Bad head localization can lead to large variation of head pose estimation, especially in the head tilt direction. Thus jointly tracking the head and estimating its pose can lead to significative head pose estimation improvements. Some researchers proposed to jointly track the head and estimates the pose. The tracking methodology we proposed in this thesis will be designed in the joint head tracking and pose estimation framework.

In this chapter, we present an appearance based joint head tracking and pose estimation algorithm embedded within a probabilistic framework. The head appearance model is based on texture features, skin color features, and binary features obtained from background subtraction. On one hand, texture features are robust to illumination condition variations while being sensitive to background clutter. On the other hand, binary mask characteristic of the presence of skin modeled with an adaptive skin color distribution, and binary features obtained from an adaptive background subtraction process make our appearance model robust to illumination changes and background clutter. In addition, a head silhouette model is built from the binary background feature, and used to detect candidate locations for the head. The tracking methodology relies on a probabilistic framework based on sequential Monte Carlo (SMC) methods. Three SMC methods are investigated:

- The first method is based on the standard importance sampling (IS), described in Section 2.2.2, and which allows for the tracking of multiple modes in the filtering distribution resulting in a better handling of ambiguities than Kalman filters.

- The second SMC method is a Rao-Blackwellized version of the IS particle filter which should results in a reduction of the required number of samples for a good tracking.
- The third SMC method is a Markov chain Monte Carlo (MCMC) method which is supposed to be more efficient than IS in high dimensional spaces.

The head pose tracking methodologies we proposed will be evaluated using part of the data of our head pose video database, IHPD, presented in Section 3.

This chapter is organized as follows. Section 4.1 presents our head appearance modeling. Section 4.2 describes our joint head location and pose tracking method with a mixed state particle filter (MSPF) based on IS. Section 4.3 describes the Rao-Blackwellisation of the MSPF. Section 4.4 presents a joint head tracking and pose estimation in a MCMC framework. Section 4.5 presents an algorithm that first track the head using a PF relying on the color histogram and head silhouette models, and then estimate the head pose from the resulting head location. This algorithm will be used for comparison with our joint head tracking and pose estimation methods. Section 4.6 presents the evaluation set-up and gives the experiments we conducted to study the performances of our tracking methods. Section 4.7 concludes the Chapter.

4.1 Head pose modeling

Head pose modeling is the preliminary step for head pose tracking. The goal of head pose modeling is to build a representation of the image appearance of heads taking into account the variation of appearance due to orientation changes. In Section 2.1 various head pose models have been presented. In this Section we present our head pose modeling.

4.1.1 Head pose modeling

We use the Pointing database [48] to build our head pose models since the discrete set of pan and tilt values available covers a larger range of poses than the one found in other databases (e.g. FERET, PIE [46, 47]). Although we use the Pointing database to build our head pose models, in tracking situation we will use the PIE head pose representation presented in Section 3.1. We use the PIE representation because it has the computational advantage that, given the appearance of a head pose $(\alpha, \beta, 0)$, the appearance corresponding to the change in the head roll (α, β, γ) can be obtained by applying an image in plane rotation of angle γ to the image appearance of the pose $(\alpha, \beta, 0)$. In the PIE representation, head roll appearance variations do not need to be modeled.

Texture and color based head pose models are built from all the sample images available for each of the 79 discrete head poses $\theta \in \Theta = \{\theta_k, k = 1, ..., 79\}$. The pointing database is composed of 93 head poses, but when passing in the PIE representation, for the profile pose ($\alpha = 90$) only one pose value needs to be retained ($\beta = 0$). The other profile pose are in-plane rotation variations of this one. Figure 4.1 shows a person in the Pointing database, at all the poses that will be used to build the appearance models. In the Pointing database, there are 15 people per pose. Ground truth image patches are obtained by locating a tight bounding box around the head. Because of the small number of people

Figure 4.1: Pointing database head poses used to build appearance models.

(a) Head image

(b) A coarse scale Gaussian filter and two Gabor filters output

(c) Face skin mask

Figure 4.2: Frontal image example and its corresponding features

in the database which will make model training quite difficult, we introduced more variability in the training set by generating virtual training images from the located head images. More precisely, new training patches were generated by applying small random perturbations to the head location and size of the original head patches.

Head pose texture model

Head pose texture is modeled with the output of three filters $\Psi_i, 1 = 1, ..., 3$: a Gaussian at coarse scale and two Gabor filters at two different scales, as illustrated in Figure 4.2(b). Training image patches are obtained by locating a tight bounding box around the head. The image patches are resized to the same resolution 64×64 and preprocessed by histogram equalization to reduce the effect of lighting conditions. Patches are then filtered by each of the above filters at subsampled pixel locations and the output of the filter are concatenated into a single feature vector. The final texture feature is a column vector, denoted by $z^{tex} = (z^{tex,1}, .., z^{tex,N_{tex}})^T$, where T denotes the transpose operator, belongs to a $N_{tex} = 552$ dimension space.

The mean $E_k^{tex} = (E_k^{tex,1}, ..., E_k^{tex,N_{tex}})^T$ of the training feature vectors associated with each head pose $\theta_k \in \Theta$ is taken to be the texture exemplar of the pose. In addition, the diagonal covariance matrix

$\sigma_k^{tex} = diag(\sigma_k^{tex,1}, ..., \sigma_k^{tex,N_{tex}})$ of the training texture features vectors of each pose θ_k are used to define the head pose likelihood models.

Texture likelihood:
The texture likelihood of an input image characterized by its extracted features z^{tex} given a head pose θ_k, is then defined by:

$$p_{tex}(z^{tex}|k) = \frac{1}{Z_k^{tex}} \exp -\lambda_k^{tex} \rho_k(z^{tex}, E_k^{tex}) \tag{4.1}$$

where ρ_k is the normalized truncated Mahalanobis distance defined as:

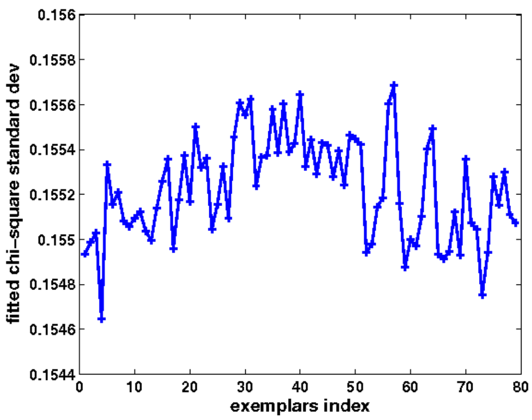
$$\rho_k(u,v) = \frac{1}{N_{tex}} \sum_{i=1}^{N_{tex}} \max\left(\left(\frac{u_i - v_i}{\sigma_k^{tex,i}}\right)^2, T_{tex}^2\right) \tag{4.2}$$

where $T_{tex} = 3$ is a threshold set to make the distance more robust to outlier components.

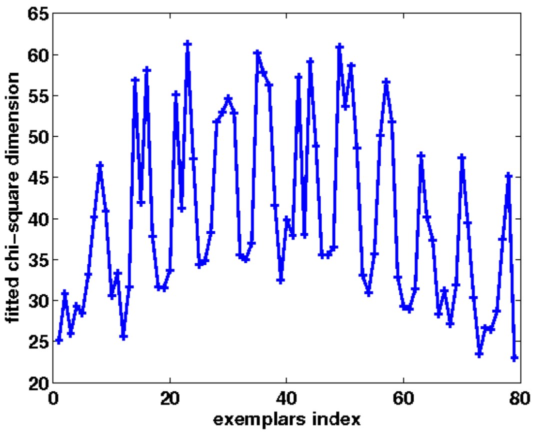
The normalization constant Z_k^{tex} and the parameter λ_k^{tex} are learned from the training data using a procedure proposed in [107]. The procedure is the following. For each head pose θ_k, the distance between the exemplar and a training features z_l^{tex}, $\rho_k(E_k^{tex}, z_l^{tex})$, is modeled as a Chi-square distribution ${s_k^{tex}}^2 \chi^2_{d_k^{tex}}$ where d_k^{tex} denotes the dimension of the fitted chi-square distribution and s_k^{tex} denotes its standard deviation. Figure 4.3 gives, for all the exemplars, the fitted Chi-square dimensions d_k^{tex} and their corresponding standard deviations s_k^{tex}. While the standard deviations of fitted Chi-square distribution to the distance between the texture exemplars and their training data are quite similar, there is a high variability between the dimensions. Recall that a Chi-square distribution with dimensionality d is a sum of d independent Gaussian variables. Hence, the variations in dimensionality of the Chi-square distributions implies that the exemplars with the higher dimensionality will be favored if no normalization is applied. Thus, the distance of all the exemplars will be fitted to a Chi-square distribution with the same dimension to normalize their values. We chose this dimension to be the average of the dimensions of the fitted Chi-square distributions. Thus, If $\bar{d}^{tex}$ denotes the mean of the chi-square dimensions, the texture likelihood parameters are given by

$$\lambda_k^{tex} = \frac{1}{2(s_k^{tex})^2} \text{ and } Z_k^{tex} = (s_k^{tex})^{\bar{d}^{tex}}. \tag{4.3}$$

When building the texture head pose models, we could have applied principal component analysis (PCA) to the input features to reduce the dimensionality of the feature space. But, as already said in Section 2.1, applying PCA to the input features would have required to keep a large number of PCA eigenvectors to precisely model the large range of head poses. Thus projecting the input features on the PCA eigenvectors would be computationally very expensive. We thus decided to use the Chi-square normalization. This normalization can be interpreted as an implicit dimensionality reduction technique because the distance in the input feature space is normalized with respect to the dimensionality of its variation using Chi-square modeling. In the texture feature case, the dimensionality of the input features is $N_{tex} = 552$ while the average of the Chi-square dimensions is $\bar{d}_{tex} = 40$.



(a) Chi-square standard deviations.



(b) Chi-square dimensions.

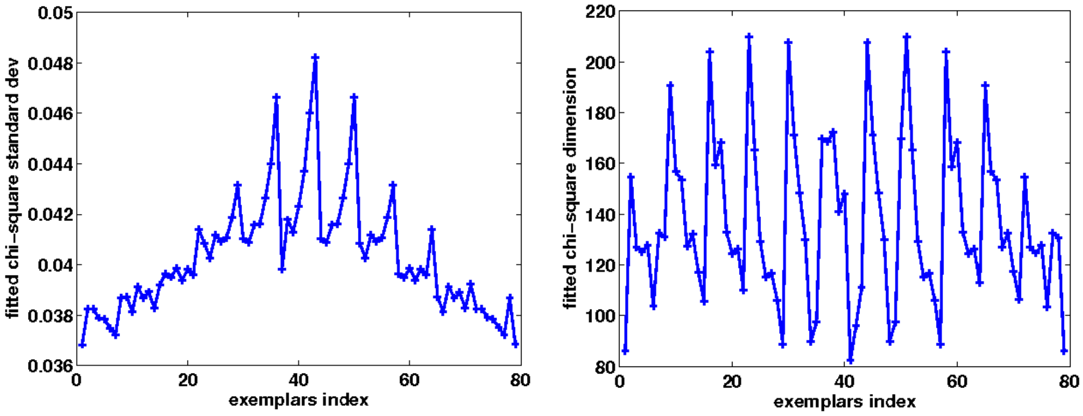
Figure 4.3: Parameters of the fitted chi-square distribution modeling the distance between each texture head pose exemplars and its training texture features vectors.

Head pose color model

To make our head models more robust to background clutter, we learned for each head pose, θ_k, a face skin color model, denoted by E_k^{skin}, using the training images belonging to the head pose. Training images are resized to 64×64, then the pixels are classified as skin or non skin to produce a binary mask having as value 0 for non-skin pixels and 1 for skin pixels, as shown in Figure 4.2(c). The skin model E_k^{skin} is taken to be the average of the training binary masks. An example of head pose skin color model is shown in Figure 4.5(b).

To detect skin pixels at run time, we model the distribution of skin pixel values with a single Gaussian distribution in the normalized (r,g) color space. As shown in [5], such a model holds well for people of any skin color tone. Thus, the parameters of a general skin color model (mean and variance), denoted by $(m_0^{skin}, \Sigma_0^{skin})$, are learned using the whole set of training images in the Pointing database. The parameters $(m_0^{skin}, \Sigma_0^{skin})$ are then adapted through time using a Maximum A Posteriori (MAP) adaptation technique, leading to the parameters $(m_t^{skin}, \Sigma_t^{skin})$ at time t. The MAP adaptation technique is described in Section 5.5. The skin color data used at time t for adaptation are computed from the image pixels extracted using the estimated mean state of the head (see Section 4.2), taking into account both the 2D spatial head localization parameters and the estimated pose, which, through the skin mask, tells which pixels of the head corresponds to the face part and can be used to collect the adaptation data.

Skin color likelihood:
The color likelihood of an input patch image with respect to the skin color model of a pose θ_k is obtained in the following way. Skin pixels are first detected on the 64×64 grid by thresholding the skin likelihood obtained using the skin color distribution model with parameters $(m_t^{skin}, \Sigma_t^{skin})$. The resulting skin



(a) Chi-square standard deviation. (b) Chi-square dimension.

Figure 4.4: Parameters of the fitted chi-square distribution modeling the distance between the head pose skin color model exemplars and their training features.

mask, z^{skin}, is then compared to the model E_k^{skin}, using the likelihood defined as:

$$p_{skin}(z^{skin}|k) = \frac{1}{Z_k^{skin}} \exp -\lambda_k^{skin} ||z^{skin} - E_k^{skin}||_1 \tag{4.4}$$

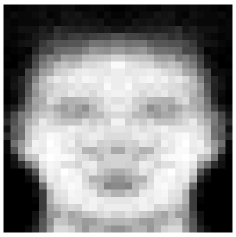
where $||.||_1$ is the normalized L^1 distance defined as

$$||u - v||_1 = \frac{1}{N_{skin}} \sum_{i=1}^{N_{skin}} |u_i - v_i| \tag{4.5}$$

with N_{skin} being the dimensionality of the skin feature vector. The normalization constants Z_k^{skin} and the parameters λ_k^{skin} are learned using the same principle as with the texture likelihood parameters. For each head pose, the distance between the skin color model and a training feature z^{skin}, $||E_k^{skin} - u^{skin}||_1$ is modeled as a Chi-square distribution $s_k^{skin^2} \chi^2_{d_k^{skin}}$. Figure 4.4 gives for all the skin color models the dimension of the fitted Chi-square distributions d_k^{skin} and their corresponding standard deviations s_k^{skin}. As in the texture modeling case, high variability can be noticed between the dimensionality of the fitted Chi-square for different poses. Normalizing the fitted Chi-square is also required, so that models with distance lying in low dimensions are not favored. If $\bar{d}^{skin}$ denotes the mean of the chi-square dimensions, the skin likelihood parameters are given by

$$\lambda_k^{skin} = \frac{1}{2(s_k^{skin})^2} \text{ and } Z_k^{skin} = s_k^{\bar{d}^{skin}}. \tag{4.6}$$

(a) Head and silhouette bounding box.



(b) Head skin model for the frontal pose.

(c) Head silhouette model.

Figure 4.5: Head modeling. Figure 4.5(a) shows the silhouette box defined as 1.5 times larger than the head box. Figure 4.5(b) shows the frontal pose head skin model. Figure 4.5(c) shows the head silhouette model. In Figure 4.5(b) and Figure 4.5(c), a bright pixel denotes a pixel likely to be a face or silhouette pixel and a dark pixel indicates a pixel unlikely to belong to the face or the silhouette of a person.

Head silhouette model

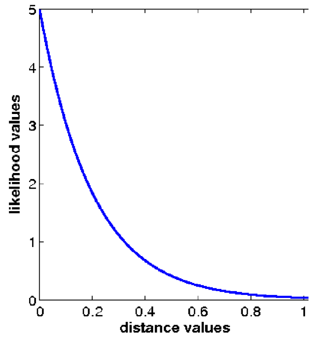
In addition to the pose dependent head models, we propose to add a head silhouette model to aid in the head localization, by taking advantage of foreground segmentation information. Figure 4.5(c) displays our head silhouette model. We built a head silhouette model, E^{sil}, by averaging head silhouette patches extracted from binary foreground segmentation images in a training set. Note that a single silhouette model is used, unlike the pose-dependent models for texture and skin color. The main reason is that at low and medium resolution, the foreground data does not provide reliable information about the pose.

Background subtraction:
The foreground objects are obtained using a background subtraction process. For each image pixel (x, y), the Hue-Saturation (HS) color is modeled as a Gaussian distribution, with parameters $(m_0^{bg}(x, y), \Sigma_0^{bg}(x, y))$ obtained from training background images. The background model is adapted using standard MAP adaptation to obtain new parameters $(m_t^{bg}(x, y), \Sigma_t^{bg}(x, y))$ at each time frame t. At each time frame, the foreground segmentation $\mathcal{F}_t$ is obtained by thresholding the likelihood of the input image I_t with respect to the background model. Therefore, the foreground is a binary mask defined as

$$\mathcal{F}_t(x, y) = \begin{cases} 1 \text{ if } \mathcal{N}(I_t(x, y); m_t^{bg}(x, y), \Sigma_t^{bg}(x, y)) < T_{bg} \\ 0 \text{ otherwise} \end{cases} \tag{4.7}$$

where T_{bg} is a threshold used to define the outliers to the background model. The resulting of the background segmentation is a binary mask $\mathcal{F}_t$ having 0 as value at background pixels and 1 at foreground

(a) Foreground segmentation from background subtraction.



(b) Silhouette likelihood: exponential distribution.

Figure 4.6: Foreground segmentation and silhouette likelihood.

pixels. Figure 4.6(a) shows a foreground image obtained from background subtraction. We can notice that due to the textured background, the result is not noisy, while being informative.

Silhouette likelihood:
The silhouette likelihood works by comparing the model, E^{sil}, to a binary image patch, z^{sil}, extracted from the foreground segmentation given the hypothesized head location. Given the spatial configuration of the head (tight bounding box around the head) the spatial configuration of the silhouette is centered at the same point but is 1.5 times bigger with the same in-plane rotation, as illustrated in Figure 4.5(a). A poor match between E^{sil} and z^{sil} indicates the presence of foreground and background pixels at unexpected locations, probably due to a poor head location hypothesis. The head silhouette likelihood is defined as an exponential distribution:

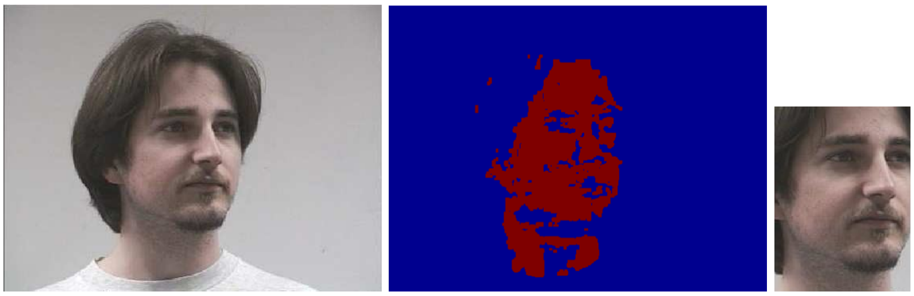
$$p_{sil}(z^{sil}) \propto \lambda_{sil} \exp - \left(\lambda_{sil} ||z^{sil} - E^{sil}||_1 \right), \tag{4.8}$$

where $||.||_1$ is the normalized L1 distance defined in Equation 4.5, and λ_{sil} is the likelihood parameter that we set to the value $\lambda_{sil} = 5$. Figure 4.6(b) shows the variation of the silhouette likelihood with respect to the distance between an input silhouette feature and the silhouette model. The likelihood is chosen to be not very peaky, because the background segmentation can be noisy sometimes.

4.1.2 Head pose modeling evaluation

We conducted two series of experiments to evaluate our head pose modeling. The first series follows the CLEAR Evaluation Workshop protocol on the Pointing database [103]. The second series is a variation of this protocol, where we avoid to have the images of the same persons in the training and test sets.

As said previously, the Pointing database is composed of 15 people, recorded at 93 poses in two sessions. In the CLEAR evaluation protocol, the data of the first session are used as training set, and data of the second session are used as test set. For head localization, two experimental conditions were considered. In the first case, the heads were localized by hand. In the second case, the head localization



(a) Example image from CLEAR evaluation data set.
(b) Detected skin area.
(c) Head localized from skin area.

Figure 4.7: Head localization with skin segmentation in CLEAR data.

was detected automatically by segmenting the skin area. A skin color model, built from the training data, was used to localize the skin area. Then, head bounding boxes were extracted from the detected skin regions. Figure 4.7 illustrates the automatic head localization process using skin color that we have used. Other researchers who participate to the CLEAR evaluation workshop used also skin color to localize the head [108, 109, 110]. Localizing heads using skin segmentation was possible because faces were the only skin-like areas in the images and there were only one face in each image. It has to be noticed that in the following experiments, the training and the test data of a given experiment are extracted using the same head extraction procedure (manual or automatic).

From the training data we built the head pose models described previously in Section 4.1. Then, for each image in the test set, we computed the likelihood of the observation, extracted using the localization procedure, with respect to each pose models. Given a test image, the texture and skin color observations z^{tex} and z^{skin} are extracted. Then three models are compared: the texture model recognition, the skin color model recognition and the texture and color fusion recognition. For the texture only model, the texture observation of the test image is compared to the texture models of the head poses using the texture likelihood defined in Equation 4.1. The recognized head pose, $\theta_{\hat{k}}$, is the pose with the highest texture likelihood:

$$\hat{k} = \text{argmax}_k \, p_{tex}(z^{tex}|k). \tag{4.9}$$

For the skin color model, a test skin color observation is compared to the skin color models and the recognized pose $\theta_{\hat{k}}$ is the pose with the highest skin color likelihood defined in Equation 4.4:

$$\hat{k} = \text{argmax}_k \, p_{skin}(z^{skin}|k). \tag{4.10}$$

In the fusion case, both the texture and the color modalities are used. The recognized head pose, $\theta_{\hat{k}}$, is the pose maximizing the likelihood:

$$\hat{k} = \text{argmax}_k \, p_{tex}(z^{tex}|k)p_{skin}(z^{skin}|k). \tag{4.11}$$

localization	by automatic skin segmentation		by hand	
error	pan	tilt	pan	tilt
texture	13.2	14.1	11.1	11.1
color	30.6	32.2	13.7	14.3
fusion	11	11.5	9.8	10.3

Table 4.1: Static head pose estimation: CLEAR evaluation setup.

method	Ba *et al*	Voit *et al* [108]	Tu *et al* [109]	Gourier *et al* [110]
pan error	11	12.3	14.1	10.3
tilt error	11.5	12.7	14.9	15.9

Table 4.2: Comparison with state of the art algorithms for head pose estimation using the CLEAR evaluation protocol: the second column gives our results (texture and skin color used jointly), the 3 last columns give results presented at the CLEAR evaluation Workshop.

The pose estimation measure is taken as the average of the absolute difference between the ground truth and the estimated pose.

Table 4.1 shows the pan and tilt estimation error results when using the texture model only, the skin color model only, and both models jointly. From the results we can conclude that the use of texture features lead to better estimate than the skin features. Besides, using texture and color together significantly improves the head pose recognition. This shows that even if skin color models give inaccurate results, fusing texture with skin color improves the head pose recognition. Table 4.1 also shows that when the head are localized by hand, especially with the skin color case, the pose error is lower giving an insight into the head pose estimation sensitiveness to head localization. Table 4.2 shows that our head pose recognition method based on the fusion of texture and skin color is competitive with respect to state-of-the-art head pose recognition methods that were presented at the CLEAR evaluation workshop 2006 [108, 109, 110, 111].

The CLEAR Evaluation set up, by using images of the same persons in both the training and test sets, does not give reliable information about the generalization ability of head pose models to recognize the poses of unseen persons. Thus, we defined a second evaluation framework with the Pointing database. More specifically, we mixed the images of the two recording sessions, and in turn left aside as test data all images of one of the person, while using images of the remaining persons as training data. Table 4.3 reports the performances of our models using this second protocol. Overall, we can see that the performance are similar to the results obtained with the CLEAR evaluation protocol. However, we can notice that while the pan errors are very close in both protocols, the tilt errors tend to be higher in the unseen setup. This indicates that the tilt estimation is more sensitive to the individual person appearance than the pan estimation.

In this section, we built a discrete representation of head poses image appearance $E_k = (E_k^{tex}, E_k^{skin}, E^{sil})$ using texture, skin color and foreground features. This representation will be used in a probabilistic framework for head pose tracking, as will be described in the remaining of this Chapter.

localization	by automatic skin segmentation		by hand	
error	pan	tilt	pan	tilt
texture	11.7	14.4	12.1	14.5
color	29.1	28	12.9	14.3
fusion	10.3	13.5	9.51	12.4

Table 4.3: Static head pose estimation: unseen person setup.

4.2 Joint head tracking and pose estimation with a mixed state particle filter (MSPF)

Particle filtering (PF) implements a recursive Bayesian filter by Monte-Carlo simulations. Let X_t represents the state at time t, and $z_{1:t}$ the sequence of observations up to time t. Furthermore, let $\{X_{t-1}^{(n)}, w_{t-1}^{(n)}\}_{n=1}^{N_s}$ denotes a set of weighted samples characterizing the pdf $p(X_{t-1}|z_{0:t-1})$, where $\{X_{t-1}^{(n)}, n = 1, \ldots, N_s\}$ is a set of support points with associated weights $w_{t-1}^{(n)}$. The goal of PF is to approximate, $p(X_t|z_{1:t})$, the current pdf of the state given the sequence of observations via sampling with a set of samples and associated weights. At each time, the samples and weights can be chosen according to the importance sampling principle (IS) described in Section 2.2.2. We make the following assumptions: the state follows a first order Markov process; and the observations are independent given the states. We also assume that we have a proposal function $q(X_t|X_{t-1}, z_t)$ that makes use of the current observation. Then, the current pdf can be approximated as:

$$p(X_t|z_{1:t}) \propto \sum_n w_{t-1}^{(n)} W_t^{(n)}(X_t) q(X_t|X_{t-1}^{(n)}, z_t) \text{ with } W_t^{(n)}(X_t) \propto \frac{p(z_t|X_t)p(X_t|X_{t-1})}{q(X_t|X_{t-1}^{(n)}, z_t)} \tag{4.12}$$

Figure 4.8 summarizes the steps of the pdf approximation with discrete particles and weights [1].

In order to implement the filter for tracking, four elements have to be defined:

1. a state model X_t describing the object;
2. a dynamical model $p(X_t|X_{t-1})$ modeling the temporal evolution of the state;
3. a proposal function $q(X_t|X_{t-1}, z_t)$ which role is to sample new state candidates in informative regions of the state space;
4. an observation model $p(z_t|X_t)$ which measures the adequacy between the observations and the state. This is an essential term where the data fusion occurs. The head pose models presented in Section 4.1 will be used to build the observation model.

In the following, we describe the four elements, necessary for the PF implementation, in the case of head pose tracking.

[1] When the PF comprises a resampling step as shown in Figure 4.12, the multiplicative term $w_{t-1}^{(n)}$ in Equation 4.12 can be discarded because after resampling, the particles are uniformly distributed.

1. **initialization step:**
   for $n = 1, ..., N_s: \ X_0^{(n)} \sim p(X_0), \ w_0^{(n)} = \frac{1}{N_s}, \ t = 1$
2. **IS step:**
   - $\tilde{X}_t^{(n)} \sim q(X_t|X_{t-1}^{(n)}, z_t), \ \tilde{w}_t^{(n)} \propto \frac{p(z_t|\tilde{X}_t^{(n)})p(\tilde{X}_t^{(n)}|X_{t-1}^{(n)})}{q(\tilde{X}_t^{(n)}|X_{t-1}^{(n)}, z_t)}, \ n = 1, ..., N_s$
   - $\tilde{w} = \sum_n \tilde{w}_t^{(n)}, \ \tilde{w}_t^{(n)} = \frac{\tilde{w}_t^{(n)}}{\tilde{w}}, \ n = 1, ..., N_s$
3. **filter output step:**
   compute the state expectation from the particles using Equations 4.33 and 4.34.
4. **selection step:**
   resample with replacement particles according to the weights: $\{\tilde{X}_t^{(n)}, \tilde{w}_t^{(n)}\} \longrightarrow \{X_t^{(n)}, w_t^{(n)} = \frac{1}{N_s}\}$;
5. **time increment step:**
   set $t = t + 1$ then go to step 2

Figure 4.8: Recursive pdf estimation with SIR algorithm.

4.2.1 State space

The MSPF approach, as proposed in [107], allows to represent jointly, in the same state variable, discrete variables and continuous variables. In our specific case, the state $X = (S, r, k)$ is the conjunction of the continuous variable S and the discrete variables r and k. For the variable $S = (x, y, s^y, e)$, $(x, y)^T$ defines the head location. s^y is a head height scale, with respect to a reference height L^y, defining the head height. e is an eccentricity variable defined by the ratio of the head width over the head height. Given the reference head width L^x and head height L^y, the head width scale s^x can be computed as:

$$s^x = e\frac{s^y L^y}{L^x} \tag{4.13}$$

The continuous variable S and the discrete variable r, together, parameterize the 2D spatial transform $\mathcal{T}_{(S,r)}$ defined as

$$\mathcal{T}_{(S,r)}u = \begin{pmatrix} s^x & 0 \\ 0 & s^y \end{pmatrix} \begin{pmatrix} \cos r & -\sin r \\ \sin r & \cos r \end{pmatrix} u + \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4.14}$$

which characterizes the object's configuration in the image plane. The vector $u = (u_x, u_y)^T$ is a point in a reference frame. The vector $(x, y)^T$ specifies the translation, i.e. the position of the object in the image plane, (s^x, s^y) denote the width and height scales of the object according to a reference size, and r specifies the in-plane rotation angle of the object. The parameter r was discretized for convenience, though this not a necessity of the approach.

The discrete variable k labels an element of the set of head pose models E_k. Together with the variable r, which defines the head roll in the PIE representation, k defines the head pose. As presented in

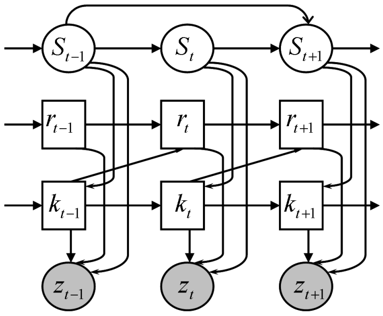
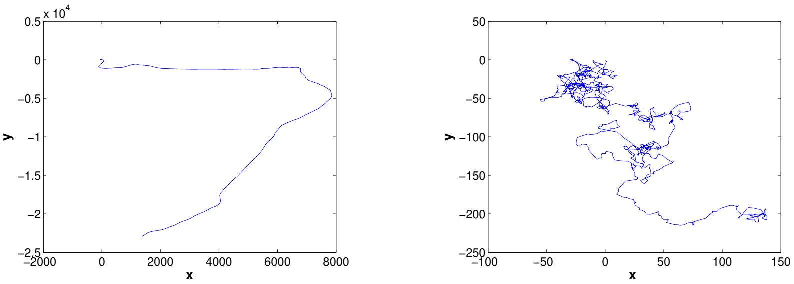


Figure 4.9: Mixed state graphical model. Continuous variables are represented by circles and discrete variables by squares. Hidden variables are bright and observation variables are shaded.



(a) Simulation of head motion with the classical motion model (ϖ=1).

(b) Simulation of head motion with the Langevin motion model (ϖ=0.5).

Figure 4.10: Simulated head trajectories. Figure 4.10(a) shows a simulation of the classical motion model with ϖ=1. Figure 4.10(b) shows a simulation of the Langevin model. For 1000 iterations, with exactly the same noise models, the trajectory obtained with a Langevin model seems more realistic.

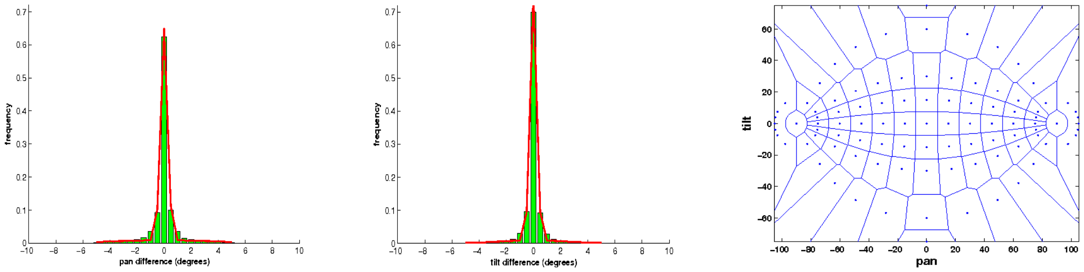
Section 3, in the PIE representation, an in-plane rotation of the head in the image plane represents a head roll angle. Thus, the exemplars variable k models the head pan-tilt angle variations, and the variable r models the head roll angle variations.

4.2.2 Dynamical models

The process density on the state sequence is modeled as a second order process. We assume that the process density can be factorized as :

$$p(X_t|X_{t-1}, X_{t-2}) = p(S_t|S_{t-1}, S_{t-2})p(k_t|k_{t-1}, S_t)p(r_t|r_{t-1}, k_{t-1}) \tag{4.15}$$

Figure 4.9 gives the graphical model of the dynamic process corresponding to this factorization. Let us now describe the three terms involved in Equation 4.15.



(a) Head pan difference distribution.

(b) Head tilt difference distribution.

(c) Voronoi segmentation in the PIE head pose space given the discrete pose.

Figure 4.11: Head pose dynamic parameters. Green: histogram of pan differences in Figure 4.11(a), and tilt differences in Figure 4.11(b). Red: fitted GMM to temporal pan and tilt differences. Figure 4.11(c) shows areas associated with each discrete pose obtained from a Voronoi segmentation.

Spatial configuration dynamic:
The dynamic of the continuous variable S_t is modeled as

$$\begin{aligned} x_t &= x_{t-1} + \varpi(x_{t-1} - x_{t-2}) + \eta_t^x \\ y_t &= y_{t-1} + \varpi(y_{t-1} - y_{t-2}) + \eta_t^y \\ e_t &= e_{t-1} + \eta_t^e \\ s_t^y &= s_{t-1}^y + \eta_t^{s^y} \end{aligned} \tag{4.16}$$

where η_t^x, η_t^y, η_t^e and $\eta_t^{s^y}$ are centered Gaussian noise models. The dynamic of the head location variable $(x_t, y_t)^T$ is modeled as Langevin dynamic [112]. In Equation 4.16, $\varpi < 1$ is the Langevin parameter damping the velocity of the object. The Langevin dynamic is well suited to model moving head because it models object's motion subject to friction. In classical head motion modeling, defined by dynamic with a parameter $\varpi = 1$, the object has the tendency to gain speed very quickly. In head modeling with a Langevin dynamic, the motion of the object is slower. Thus, when modeling head motion, the exploration of the space is more efficient with a Langevin dynamic, as illustrated in Figure 4.10.

Head pose dynamic:
The dynamic of the discrete pose variable k_t is defined by the transition process $p(k_t|k_{t-1}, S_t)$ defined as

$$p(k_t|k_{t-1}, S_t) \propto p_0(\theta_{k_t})p(k_t|k_{t-1})p(k_t|S_t) \tag{4.17}$$

where $p_0(\theta_{k_t})$ is a prior on the head pose, $p(k_t|k_{t-1})$ models transition between head poses, and $p(k_t|S_t)$ models a prior on the head pose given the head spatial configuration.

The prior on the head pose $p_0(\theta_{k_t})$ defined as:

$$p_0(\theta_{k_t}) = (\alpha_{k_t}, \beta_{k_t}, \gamma_{k_t})) = \mathcal{N}(\alpha_{k_t}, 0, \sigma_\alpha)\mathcal{N}(\beta_{k_t}, 0, \sigma_\beta) \tag{4.18}$$

is a coarse Gaussian distribution centered at the frontal pose $(\alpha, \beta) = (0, 0)$ with standard deviation $(\sigma_\alpha, \sigma_\beta) = (45, 30)$. This prior is used to slightly favor near frontal poses with respect to near profile poses. It compensates for the fact that we observed that it is easier to fit (i.e. obtain higher likelihood) a profile than a frontal head poses.

The transition term between poses, $p(k_t|k_{t-1})$, is learned from continuous head pose ground truth. First, we assume that for each pose $\theta = (\alpha, \beta, \gamma)$ the pan component α and the tilt component β are independent leading to

$$p(\theta_t|\theta_{t-1}) = p(\alpha_t|\alpha_{t-1})p(\beta_t|\beta_{t-1}). \tag{4.19}$$

The continuous transition between the head pose pan and tilt are modeled using the absolute difference between head pose components $p(\alpha_t|\alpha_{t-1}) = p(\alpha_t - \alpha_{t-1})$ and $p(\beta_t, \beta_{t-1}) = p(\beta_t - \beta_{t-1})$. The temporal differences are used to model the dynamics because it allows to estimate the transition distribution in all part of the pose space even when we do not have data.

The temporal pan and tilt differences $\alpha_t - \alpha_{t-1}$ and $\beta_t - \beta_{t-1}$ are modeled as two Gaussian mixture models (GMM) p_α and p_β in the continuous space. The mixtures comprises two Gaussian centered on 0. Intuitively, one mixture component is expected to model situations when the head remains static, while the other one will account for head pose variations when the person changes his head pose. Figures 4.11(a) and 4.11(b) displays the fitted mixtures.

The pose variable k_t is discrete. The GMM learned to modeled the transition are continuous distributions and need to be adequately discretize. The discretization is important because in the PIE representation, the discrete poses do not represent a uniform sampling of the pose space, as can be seen in Figure 4.11(c). Each discrete pose represents an area of the space of poses. Because the discretization is not uniform, the areas do not have the same surface. Thus, the transition between the discrete poses have to take into account the areas they represent. Therefore, we need to define transitions betweens regions of the pose space. The continuous GMM are used to compute the transition between two regions of the head pose space, $\mathcal{O}_i$ and $\mathcal{O}_j$, as:

$$\begin{aligned} p(\mathcal{O}_j|\mathcal{O}_i) &= \frac{p(\mathcal{O}_j, \mathcal{O}_i)}{p(\mathcal{O}_i)} && (4.20) \\ &= \frac{\int_{\mathcal{O}_i}\int_{\mathcal{O}_j} p(\theta_j|\theta_i)p_0(\theta_i)d\theta_j d\theta_i}{\int_{\mathcal{O}_i} p_0(\theta_i)d\theta_i}. && (4.21) \end{aligned}$$

In our case, the integrals in (4.20) are computed by discretizing the head pose space. The transition process between the discretized poses k_{t-1} and k_t is defined from the transition between regions as $p(k_t|k_{t-1}) = p(\mathcal{O}_{k_t}|\mathcal{O}_{k_{t-1}})$, where $\mathcal{O}_k$ is the region represented by the discrete head pose θ_k according to a Voronoi segmentation of the head pose space given the discretization. Figure 4.11(c) gives the segmentation of the PIE head pose space into the Voronoi areas.

Finally, the priors on the head pose given the head spatial configuration $p(k_t|S_t)$ is defined as:

$$p(k_t|S_t) = \frac{p^e_{k_t}(e_t)}{\sum_{k'_t} p^e_{k'_t}(e_t)} \tag{4.22}$$

where $p^e_{k_t}$ is a Gaussian prior on the head eccentricity learned from the training data. This term is used in the dynamic model to take into account information contained into the head spatial configuration for the pose evolution.

Head roll dynamic:
Finally, $p(r_t|r_{t-1}, k_{t-1})$, the dynamic of the in plane rotation variable, is also learned using the sequences in the training data set, and comprises a Gaussian prior on the head roll $p_\Theta(r_t)$. More specifically, the PIE pan and tilt space is divided into nine regions $\Theta_i,\ i = 1, ..., 9$: obtained by equally dividing the pan and tilt range (from -90 to 90 degrees) with a 60 degrees step. Inside each region Θ_i, a Gaussian distribution $\eta^{r,i} = p(r_t - r_{t-1})$ is fitted to the roll temporal differences $r_t - r_{t-1}$ of the training data which head pose falls the regions in Θ_i. Thus, inside each region Θ_i the transition between roll values is modeled as:

$$p_{r,i}(r_t|r_{t-1}) = \mathcal{N}(r_t; r_{t-1}, \sigma^{r,i}) \tag{4.23}$$

where $\sigma^{r,i}$ is the standard deviation of the distribution $\eta^{r,i}$. The transition between roll values for pan and tilt belonging to the region Θ_i can also be written

$$r_t = r_{t-1} + \eta^{r,i} \tag{4.24}$$

A prior distribution on the roll values $p_{\Theta_i}(r)$ is also learned by fitting a Gaussian distribution to the roll values of head pose with pan-tilt values in Θ_i. If we define

$$\Phi : \theta \rightarrow i \tag{4.25}$$

to be the mapping between the pan-tilt space to the indices of the 9 regions, the roll transition is defined as

$$p(r_t|r_{t-1}, k_{t-1}) \propto p_{r,\Phi(k_{t-1})}(r_t|r_{t-1}) p_{\Theta_{\Phi(k_{t-1})}}(r_t). \tag{4.26}$$

Hence, the variable k_{t-1} acts on the roll dynamic like a switching variable, and this also holds for the prior on the roll value. Finally, the transition between the discrete roll values is obtained by just discretizing the continuous transition distributions. This is possible because the roll discretization is uniform.

4.2.3 Proposal function

The proposal function $q(X_t|X_{t-1}, z_t)$ is used to sample new state X_t using the previous state X_{t-1} and knowledge contained in the current observation z_t. More clearly, the proposal function is defined as:

$$q(X_t|X_{t-1}, z_t) = (1 - \varepsilon) p(X_t|X_{t-1}, X_{t-2}) + \frac{\varepsilon}{N_d} \sum_{j=1}^{N_d} p(X_t|\check{X}_j(z_t)) \tag{4.27}$$

where $p(X_t|X_{t-1}, X_{t-2})$ is the object dynamical model defined in Section 4.2.2, ε is a mixture weight defining the proportion of state to be sampled from detection, $\check{X}_j(z_t)$ are head states obtained by detecting heads location from the background subtracted image using the head silhouette model and N_d is the number of detected heads. This form of proposal allows to recover from tracking failure as it em-

(a) Foreground segmentation from background subtraction.

(b) Spatial configuration candidates (green boxes) obtained by scanning the foreground image with the head silhouette model.

Figure 4.12: Candidates spatial configuration obtained from head detection using the foreground image and the head silhouette model.

beds a re-initializing component $\frac{1}{N_d}\sum_{j=1}^{N_d} p(X_t|\check{X}_j(z_t))$ sampling new states around detected heads[2]. A detected head

$$\check{X}_j(z_t) = (\check{S}_j(z_t), \check{r}_j(z_t), \pi_{z_t}(k)) \tag{4.28}$$

is defined by a spatial configuration, $\check{S}_j = (\check{x}_j, \check{y}_j, \check{e}_j, \check{s}_j^y)$ defining the location and size of the detected head (see Figure 4.12(b)), a head roll $\check{r}_j = 0$, and a distribution on the head pose $\pi_{z_t}(k)$ defined as

$$\pi_{z_t}(k) \propto p_{tex}(z_t^{tex}(\check{S}_j, \check{r}_j)|k) p_{skin}(z_t^{skin}(\check{S}_j, \check{r}_j)|k) \tag{4.29}$$

according to which a new head pose variable $\check{k}_j$ can be sampled. The distribution $p(X_t|\check{X}_j(z_t))$ assumes that the variables are independent and is modeled as:

$$\begin{aligned} x_t &= \check{x}_j + \eta^x \\ y_t &= \check{y}_j + \eta^y \\ e_t &= \check{e}_j + \eta^e \\ s_t^y &= \check{s}_j^y + \eta^{s^y} \\ r_t &= \check{r}_j + \eta^{r,\Phi(\check{k}_j)} \end{aligned} \tag{4.30}$$

The detected head locations $\check{S}_j(z_t)$ are obtained by comparing binary foreground feature $z_t^{sil}(S_l, r)$ extracted from spatial configuration $(S_l, r = 0)$ defined by uniformly subsampled image locations and a range of head sizes. The detected head locations are the configurations S_l having their silhouette likelihood higher than T_{sil}, a detection threshold:

$$\{\check{S}_j(z_t),\ j = 1, ..., N_d\} = \{S_l / \, p_{sil}(z_t^{sil}(S_l, r)) > T_{sil}\}. \tag{4.31}$$

[2]In case no head is detected, $N_d = 0$, only the dynamic $p(X_t|X_{t-1})$ is used for sampling ($\varepsilon = 0$).

4.2.4 Observation models

The observation likelihood $p(z|X)$ is defined as follows :

$$p(z|X = (S, r, k)) = p_{tex}(z^{tex}(S, r)|k)p_{skin}(z^{skin}(S, r)|k)p_{sil}(z^{sil}(S, r)) \tag{4.32}$$

where the observations $z = (z^{tex}, z^{skin}, z^{sil})^T$ are composed of texture, skin color, and binary foreground observations. Also, we assumed that the observations are conditionally independent given the state. The texture likelihood p_{tex}, the skin observation likelihood p_{skin} and the silhouette likelihood p_{sil} are defined in Section 4.1 in Equations 4.1, 4.4, and 4.8, respectively.

The computation of the observations is done as follows. First the image patch associated with the image spatial configuration of the state, (S, r), is cropped from the image according to $\mathcal{C}(S, r) = \{\mathcal{T}_{(S,r)}u,\ u \in \mathcal{C}\}$, where $\mathcal{C}$ corresponds to the set of 64x64 locations defined in a reference frame. Then, the texture, the skin color and the binary observations are computed using the procedure described in Section 4.1.

4.2.5 Filter output

We need to define what we use as output of our filter. The set of particles defines a pdf over the state space. Thus, we can use as output the expectation value of this pdf, obtained by standard averaging over the particle set. Note that usually, with mixed-state PFs, averaging over discrete variable is not possible (e.g. if a discrete index represents a person identity). However, in our case, there is no problem since our discrete indices correspond to real Euler angles, in the PIE representation, which can be combined. Given the set of particles $\{X_t^{(n)} = (S_t^{(n)}, r_t^{(n)}, k_t^{(n)}), w_t^{(n)},\ n = 1, ..., N_s\}$ the spatial configuration and the in plane rotation output of the filter, $\hat{S}_t$ and $\hat{r}_t$, are defined as:

$$\begin{aligned} \hat{S}_t &= \sum_{n=1}^{N_s} w_t^{(n)} S_t^{(n)} \\ \hat{r}_t &= \sum_{n=1}^{N_s} w_t^{(n)} r_t^{(n)} \end{aligned} \tag{4.33}$$

The head pose output of the filter, $\hat{\theta}_t = (\hat{\alpha}_t, \hat{\beta}_t, \hat{\gamma}_t)$, is defined as:

$$\begin{aligned} \hat{\alpha}_t &= \sum_{n=1}^{N_s} w_t^{(n)} \alpha_{k_t^{(n)}} \\ \hat{\beta}_t &= \sum_{n=1}^{N_s} w_t^{(n)} \beta_{k_t^{(n)}} \\ \hat{\gamma}_t &= \sum_{n=1}^{N_s} w_t^{(n)} (\gamma_{k_t^{(n)}} + r_t^{(n)}) \qquad (4.34) \end{aligned}$$

$$(4.35)$$

where the pose represented by the pose variable $k_t^{(n)}$ is denoted $\theta_{k_t^{(n)}} = (\alpha_{k_t^{(n)}}, \beta_{k_t^{(n)}}, \gamma_{k_t^{(n)}})$. It has to be noticed that, to estimate the head roll $\hat{\gamma}_t$, the particles' in-plane rotation values $r_t^{(n)}$ are taken into account.

4.3 Head pose tracking with a Rao-Blackwellize MSPF

Rao-Blackwellization can be applied when the filtering pdf of some state model variables can exactly be computed given the samples of the remaining variables. As the exemplar label k_t is discrete and belongs to a finite set, it fulfils the conditions for an exact pdf computation.

Given the graphical model of our filter in Figure 4.9, the RBPF consists in applying the standard IS algorithm over the tracking variables S_t and r_t while applying an exact filtering step over the exemplar variable k_t, *given the samples of the tracking variables*. For this purpose, the pdf can be written:

$$p(S_{1:t}, r_{1:t}, k_{1:t}|z_{1:t}) = p(k_{1:t}|S_{1:t}, r_{1:t}, z_{1:t})p(S_{1:t}, r_{1:t}|z_{1:t}) \tag{4.36}$$

In practice, only the sufficient statistics $p(k_t|S_{1:t}, r_{1:t}, z_{1:t})$ needs to be computed. The pdf $p(S_{1:t}, r_{1:t}|z_{1:t})$ is approximated via IS. Thus, in the RBPF modeling, the pdf in Equation 4.36 is represented by a set of particles

$$\{S_{1:t}^{(n)}, r_{1:t}^{(n)}, \pi_t^{(n)}(k_t), w_t^{(n)}\}_{n=1}^{N_s} \tag{4.37}$$

where

$$\pi_t^{(n)}(k_t) = p(k_t|S_{1:t}^{(n)}, r_{1:t}^{(n)}, z_{1:t}) \tag{4.38}$$

is the pdf of the exemplars given a particle and a sequence of measurements, and

$$w_t^{(n)} \propto p(S_{1:t}^{(n)}, r_{1:t}^{(n)}|z_{1:t}) \tag{4.39}$$

is the weight of the tracking state particle. In the following, we detail the methodology to derive the exact steps to compute $\pi_t^{(n)}(k_t)$ and the SIS steps to compute the particles weights $w_t^{(n)}$.

4.3.1 Deriving the exact step

The goal here is do derive $p(k_t|S_{1:t}, r_{1:t}, z_{1:t})$. As k_t is discrete, this can be done using prediction and update steps similar to those involved in Hidden Markov Model (HMM) [113].

Prediction step for variable k_t:
Given the new samples of S and r at time t, $p(k_t|S_{1:t}, r_{1:t}, z_{1:t-1})$ the prediction distribution of k_t can be evaluated as:

$$\begin{aligned} p(k_t|S_{1:t}, r_{1:t}, z_{1:t-1}) &= \sum_{k_{t-1}} p(k_t, k_{t-1}|S_{1:t}, r_{1:t}, z_{1:t-1}) \\ &= \sum_{k_{t-1}} p(k_t|k_{t-1}, S_t)p(k_{t-1}|S_{1:t}, r_{1:t}, z_{1:t-1}) \end{aligned} \tag{4.40}$$

The first term, $p(k_t|k_{t-1}, S_t)$, is the head pose dynamic defined in Section 4.2.2 by Equation 4.17. However, unlike in the standard RBPF, the second term $p(k_{t-1}|S_{1:t}, r_{1:t}, z_{1:t-1})$, due to the extra dependency between r_t and k_{t-1}, is not equal to $p(k_{t-1}|S_{1:t-1}, r_{1:t-1}, z_{1:t-1})$. However, this term can still be computed. Exploiting the dependency assumptions of the graphical model, we have:

$$p(k_{t-1}|S_{1:t}, r_{1:t}, z_{1:t-1}) = \frac{p(r_t|r_{t-1}, k_{t-1})p(S_t|S_{t-1}, S_{t-2})p(k_{t-1}|S_{1:t-1}, r_{1:t-1}, z_{1:t-1})}{Z_1(S_t, r_t)} \quad (4.41)$$

where $p(S_t|S_{t-1}, S_{t-2})$ is the spatial configuration dynamics defined in Section 4.2.2 by Equation 4.16 and $p(r_t|r_{t-1}, k_{t-1})$ in the in-plane rotation dynamics defined in Section 4.2.2 by Equation 4.26. The normalization constant of the denominator $Z_1(S_t, r_t) = p(S_t, r_t|S_{1:t-1}, r_{1:t-1}, z_{1:t-1})$ can easily be computed by summing the numerator w.r.t. k_{t-1}:

$$Z_1(S_t, r_t) = p(S_t|S_{t-1}, S_{t-2}) \sum_{k_{t-1}} p(r_t|r_{t-1}, k_{t-1})p(k_{t-1}|S_{1:t-1}, r_{1:t-1}, z_{1:t-1}) \quad (4.42)$$

Update step for variable k_t:
When new observations z_t are available, the prediction can be updated to obtain the targeted pdf:

$$p(k_t|S_{1:t}, r_{1:t}, z_{1:t}) = \frac{p(z_t|S_t, r_t, k_t)p(k_t|S_{1:t}, r_{1:t}, z_{1:t-1})}{Z_2} \quad (4.43)$$

where the normalization constant, $Z_2 = p(z_t|S_{1:t}, r_{1:t}, z_{1:t-1})$, can be obtained by summing the numerator with respect to k_t :

$$Z_2 = \sum_{k_t} p(z_t|S_t, r_t, k_t)p(k_t|S_{1:t}, r_{1:t}, z_{1:t-1}). \quad (4.44)$$

4.3.2 Deriving the SIS PF steps

The pdf $p(S_{1:t}, r_{1:t}|z_{1:t})$ is approximated using particles whose weight is recursively computed using the standard SIS principle. The target pdf $p(S_{1:t}, r_{1:t}|z_{1:t})$ which can be written:

$$p(S_{1:t}, r_{1:t}|z_{1:t}) = \frac{p(z_t|S_{1:t}, r_{1:t}, z_{1:t-1})p(S_{1:t}, r_{1:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \quad (4.45)$$

$$= \frac{p(z_t|S_{1:t}, r_{1:t}, z_{1:t-1})p(S_t, r_t|S_{1:t-1}, r_{1:t-1}, z_{1:t-1})p(S_{1:t-1}, r_{1:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})}. \quad (4.46)$$

Using the discrete approximation of the pdf at time $t-1$, with the set of particles and weights

$$p(S_{1:t-1}, r_{1:t-}|z_{1:t-1}) \approx \sum_{n=1}^{N_s} w_{t-1}^{(n)} \delta_{S_{1:t-1}^{(n)}}(S_{1:t-1}) \delta_{r_{1:t-1}^{(n)}}(r_{1:t-1}) \quad (4.47)$$

the target pdf, $p(S_{1:t}, r_{1:t}|z_{1:t})$, can be approximated, up to the proportionality constant $p(z_t|z_{1:t-1})$, as

$$p(S_{1:t}, r_{1:t}|z_{1:t}) \propto \sum_{n=1}^{N_s} w_{t-1}^{(n)} p(z_t|S_{1:t-1}^{(n)}, S_t, r_{1:t-1}^{(n)}, r_t, z_{1:t-1}) p(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1}). \quad (4.48)$$

Which, using a proposal density $q(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t})$, can be written as:

$$p(S_{1:t}, r_{1:t}|z_{1:t}) \propto \sum_{n=1}^{N_s} w_{t-1}^{(n)} W_t^{(n)}(S_t, r_t) q(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t}) \quad (4.49)$$

where the term $W_t^{(n)}(S_t, r_t)$ is defined as:

$$W_t^{(n)}(S_t, r_t) = \frac{p(z_t|S_{1:t-1}^{(n)}, S_t, r_{1:t-1}^{(n)}, r_t, z_{1:t-1}) p(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1})}{q(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t})}. \quad (4.50)$$

The term $W_t^{(n)}(S_t, r_t)$ is defined by three components: a likelihood component

$$p(z_t|S_{1:t-1}^{(n)}, S_t, r_{1:t-1}^{(n)}, r_t, z_{1:t-1}), \quad (4.51)$$

a dynamic component

$$p(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1}), \quad (4.52)$$

and a proposal density component

$$q(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t}). \quad (4.53)$$

The likelihood component, in (4.51), is exactly the normalization constant Z_2 in (4.44). It can be readily computed when new samples are available. The dynamic component, in (4.52), can be rewritten, using the dynamic on the spatial configuration $p(S_t|S_{t-1}, S_{t-2})$, the dynamic on the head in-plane rotation $p(r_t|r_{t-1}, k_{t-1})$ and the exact distribution of the head pose variable at the previous time step $\pi_{t-1}^{(n)}$, as follows:

$$\begin{aligned} p(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1}) &= p(S_t|S_{t-1}, S_{t-2}) p(r_t|S_t, S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1}) \\ &= p(S_t|S_{t-1}, S_{t-2}) \sum_{k_{t-1}} p(r_t, k_{t-1}|S_t, S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1}) \\ &= p(S_t|S_{t-1}, S_{t-2}) \sum_{k_{t-1}} \pi_{t-1}^{(n)}(k_{t-1}) p(r_t|r_{t-1}, k_{t-1}). \quad (4.54) \end{aligned}$$

The proposal density term in (4.53) is used to sample new particles, and is defined as

$$q(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t}) = (1-\varepsilon) p(S_t, r_t|S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1}) + \frac{\varepsilon}{N_d} \sum_{j=1}^{N_d} p(S_t, r_t|\check{X}_j(z_t)) \quad (4.55)$$

where the first term is the dynamic component. The second term

$$\frac{\varepsilon}{N_d}\sum_{j=1}^{N_d} p(S_t, r_t|\check{X}_j(z_t)) \tag{4.56}$$

is the detection proposal defined in (4.27). This term is used to proposed particles around head configurations obtained from head detection using background subtraction and the silhouette model. N_d is the number of detected head and each $\check{X}_j(z_t)$ is a detected head state defined by a head location, an in-plane rotation, and a distribution on the head pose variable. Section 4.2.3 give details about the head detection process.

New samples, $(S_t^{(n)}, r_t^{(n)})$, are drawn from the proposal density and there weights [3] are computed as:

$$w_t^{(n)} \propto w_{t-1}^{(n)} W_t^{(n)}(S_t^{(n)}, r_t^{(n)}). \tag{4.57}$$

Figure 4.13 summarizes the steps of the RBPF algorithm with the additional resampling step to avoid sampling degeneracy.

4.3.3 RBPF output

At each time step, the filter outputs the mean head pose configuration. For instance, it can be obtained by first computing the head pose mean of each particle, which is given by the average of the exemplars head pose k_t with respect to the distribution $\pi_t^{(n)}(k_t)$. Then the particle head poses are averaged with respect to the distribution of the weights $w_t^{(n)}$ to give the head pose output of the RBPF. The formulae giving the RBPF spatial configuration and in-plane rotation outputs, $\hat{S}_t$ and $\hat{r}_t$, are the following:

$$\begin{aligned} \hat{S}_t &= \sum_{n=1}^{N_s} w_t^{(n)} S_t^{(n)} \\ \hat{r}_t &= \sum_{n=1}^{N_s} w_t^{(n)} r_t^{(n)}. \end{aligned} \tag{4.58}$$

[3] When a resampling step of the particles, according to their weights, is involved in the RBPF algorithm, the term $w_{t-1}^{(n)}$, in Equation 4.57, disappears in the weight computation because after resampling the remaining particles are uniformly distributed ($w_{t-1}^{(n)} = \frac{1}{N_s}$).

1. **initialization step:**
$\forall\, i$, sample $(S_0^{(n)}, r_0^{(n)})$ from $p(S_0, r_0)$, and set $\pi_0^{(n)}(.)$ to uniform and $t = 1$
2. **prediction of new head location configurations:**
sample $\tilde{S}_t^{(n)}$ and $\tilde{r}_t^{(n)}$ from the mixture
$$(\tilde{S}_t^{(n)}, \tilde{r}_t^{(n)}) \sim q(S_t, r_t | S_{1:t-1}^{(n)}, r_{1:t-1}^{(n)}, z_{1:t-1})(\text{cf Equation 4.55})$$
3. **head poses distribution of the particles:**
compute $\tilde{\pi}_t^{(n)}(k_t) = p(k_t | S_{1:t}^{(n)}, r_{1:t}^{(n)}, z_{1:t})$ using Equations 4.40 and 4.43 for all i and k_t
4. **particles weights:**
for all i compute the weights $w_t^{(n)}$ (see Equation 4.57)
5. **selection step:**
resample N_s particle $\{S_t^{(n)}, r_t^{(n)}, \pi_t^{(n)}(.), w_t^{(n)} = \frac{1}{N_s}\}$ from the set $\{\tilde{S}_t^{(n)}, \tilde{r}_t^{(n)}, \tilde{\pi}_t^{(n)}(.), \tilde{w}_t^{(n)}\}$, set $t = t + 1$ go back to step 2

Figure 4.13: RBPF Algorithm.

The head pose output of the filter $\hat{\theta}_t = (\hat{\alpha}_t, \hat{\beta}_t, \hat{\gamma}_t)$ is obtained as:

$$\begin{aligned}
\hat{\alpha}_t &= \sum_{n=1}^{N_s} w_t^{(n)} \left(\sum_k \pi_t^{(n)}(k) \alpha_k \right) \\
\hat{\beta}_t &= \sum_{n=1}^{N_s} w_t^{(n)} \left(\sum_k \pi_t^{(n)}(k) \beta_k \right) \\
\hat{\gamma}_t &= \sum_{n=1}^{N_s} w_t^{(n)} \left(\left(\sum_k \pi_t^{(n)}(k) \gamma_k \right) + r_t^{(n)} \right).
\end{aligned} \tag{4.59}$$

As a reminder, the head pose represented by the discrete pose variable k is denoted $\theta_k = (\alpha_k, \beta_k, \gamma_k)$.

4.4 Head pose tracking with an MCMC method

The MSPF or RBPF are based on the IS formalism. Choosing the proposal function for IS is sensitive. In IS, the proposal distribution has, in one throw, to propose a set of particles representing well the target distribution. Thus, in IS, the proposal function has to be similar to the target distribution. MCMC sampling methods, such as Metropolis-Hastings or Gibbs sampling, do not have this requirement. MCMC sampling methods approximate a target distribution by a chain of states. The elements of the chain are built one by one, based on a random walk process. The principle of MCMC sampling is the following: given a target distribution $p(X)$ that can be evaluated up to a proportionality constant and a proposal density $Q(X)$, build a chain of states $X^{(n)}$ which is discrete approximation of the target distribution

$p(X)$. Given the current element of the chain $X^{(n)}$, a candidate element for the chain,X^*, is generated from the proposal density. The decision to accept the new state is taken based on the quantity a defined as:

$$a = \frac{p(X^*)Q(X^{(n)})}{p(X^{(n)})Q(X^*)}. \tag{4.60}$$

The quantity a is called acceptance ratio. The proposed state is accepted with a probability a. If $u_a \sim \mathcal{U}_{[0,1]}$ is a random variable sampled from a uniform distribution on the interval $[0, 1]$, the acceptance rule for a candidate state is the following:

$$\begin{cases} X^{i+1} = X^* \; \textit{if} \; a \geq 1 \text{ or } u_a < a \\ X^{i+1} = X^{(n)} \; \textit{otherwise} \; . \end{cases} \tag{4.61}$$

In a tracking situation, the MCMC principle is applied to estimate $p(X_t|z_{1:t})$, the filtering distribution of the state X_t given the sequence of observations $z_{1:t}$. Given the Markov chain of samples representing the pdf at the previous time, $\{X_{t-1}^{(n)}, n = 1, ..., N_s\}$, a discrete approximation of the current pdf is given by:

$$\begin{aligned} p(X_t|z_{1:t}) &= \frac{p(z_t|X_t)p(X_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &\propto p(z_t|X_t) \sum_{j=1}^{N_s} p(X_t|X_{t-1}^j, X_{t-2}^j). \end{aligned} \tag{4.62}$$

Using the discrete representation of the pdf, in Equation4.62, the ratio to accept a candidate configuration, X^*, can be written as :

$$a = \frac{p(z_t|X^*) \left(\sum_{j=1}^{N_s} p(X^*|X_{t-1}^j, X_{t-2}^j)\right) Q(X_t^{(n)})}{p(z_t|X_t^{(n)}) \left(\sum_{j=1}^{N_s} p(X_t^{(n)}|X_{t-1}^j, X_{t-2}^j)\right) Q(X^*)}. \tag{4.63}$$

It is important to notice that, at each time step t, a procedure to initialize the chain that defines X_t^1, has to be defined. A possibility is to sample uniformly one of the elements of the chain at the previous time. Other possibilities are to initialize from the mean or the mode of the chain at the previous time. Figure 4.14 summarizes the different steps an MCMC algorithm.

As in the basic IS, the choice of a good proposal density is essential for the success of the algorithm. In the following we discuss possible choices of proposal density $Q(X_t)$ and their implications on the acceptance ratio.

4.4.1 Proposing new states from the previous time step samples

When defining a proposal density we could choose a density proposing states from the elements of the chain at previous step. We consider two situations, first the proposal sample from the dynamical model, secondly the proposal function use a mixture between the dynamical model and head detector.

Sampling using the dynamic:
A first possible choice is to define the proposal density as the dynamic:

$$Q(X_t) = \frac{1}{N_s} \sum_{j=1}^{N_s} p(X_t | X_{t-1}^j, X_{t-2}^j). \tag{4.64}$$

As for the IS particle filter, this choice of proposal density has a computational advantage. The summation terms in the acceptance ratio cancel out:

$$a = \frac{p(z_t|X^*) \left(\sum_{j=1}^{N_s} p(X^*|X_{t-1}^j, X_{t-2}^j)\right) \left(\sum_{j=1}^{N_s} p(X_t^{(n)}|X_{t-1}^j, X_{t-2}^j)\right)}{p(z_t|X_t^{(n)}) \left(\sum_{j=1}^{N_s} p(X_t^{(n)}|X_{t-1}^j, X_{t-2}^j)\right) \left(\sum_{j=1}^{N_s} p(X^*|X_{t-1}^j, X_{t-2}^j)\right)}, \tag{4.65}$$

resulting in a gain of computational cost. The only remaining term in the acceptance ratio is the likelihoods ratio

$$a = \frac{p(z_t|X^*)}{p(z_t|X_t^{(n)})}. \tag{4.66}$$

As a consequence, the behavior of the MCMC algorithm in this choice of proposal function will be very similar to the IS particle filter with a resampling step. The major drawback for sampling from the dynamic is that it uses no knowledge from the current observation.

Sampling using knowledge from the observations:
To sample using knowledge of the current observation we could exploit of the proposal density defined in Equation 4.27 and set the MCMC proposal to be:

$$Q(X_t) = \frac{1}{N_s} \sum_{j=1}^{N_s} q(X_t | X_{t-1}^j, z_t). \tag{4.67}$$

This proposal density is a mixture between the state dynamic model and Gaussian centered at configurations obtained from a detector applied to the current image frame. With this proposal function, the acceptance ratio becomes:

$$a = \frac{p(z_t|X^*) \left(\sum_{j=1}^{N_s} p(X^*|X_{t-1}^j, X_{t-2}^j)\right) \left(\sum_{j=1}^{N_s} q(X_t^{(n)}|X_{t-1}^j, z_t)\right)}{p(z_t|X_t^{(n)}) \left(\sum_{j=1}^{N_s} p(X_t^{(n)}|X_{t-1}^j, X_{t-2}^j)\right) \left(\sum_{j=1}^{N_s} q(X^*|X_{t-1}^j, z_t)\right)}. \tag{4.68}$$

As can be seen, two summations on the particles' set need to be done, at the numerator and denominator, resulting in a huge computational cost.

4.4.2 Sampling from the current state

In MCMC sampling, there is no particular need to sample from the elements of the chain at the previous time. New hypotheses, X^*, can be proposed based on the current element of the chain $X_t^{(n)}$. When proposing sample based on the current element of the chain, the goal is to generate random pathes which search the mode of the filtering distribution. This can be done locally by defining the proposal as a random walk

$$Q(X_t) = p(X_t|X_t^{(n)}). \tag{4.69}$$

where, reminding that in the case of head pose tracking the state variable is $X_t = (S_t, r_t, k_t)$, $p(X_t|X_t^{(n)})$ is defined as:

$$p(X_t|X_t^{(n)}) = p(S_t|S_{t-1}^{(n)})p(r_t|r_{t-1}^{(n)}, k_{t-1}^{(n)})p(k_t|k_{t-1}^{(n)}, S_t). \tag{4.70}$$

The dynamic of the head location and size variable $p(S_t|S_{t-1}^{(n)})$ is a taken to be:

$$\begin{array}{l} x_t = x_{t-1} + \eta_t^x \\ y_t = y_{t-1} + \eta_t^y \\ e_t = e_{t-1} + \eta_t^e \\ s_t^y = s_{t-1}^y + \eta_t^{s^y} \end{array} \tag{4.71}$$

where, as a reminder, $S_t = (x_t, y_t, e_t, s_t^y)$, and η_t^x, η_t^y, η_t^e and $\eta_t^{s^y}$ are centered Gaussian noise models. The dynamic of the head in-plane rotation, $p(r_t|r_{t-1}^{(n)}, k_{t-1}^{(n)})$, and the head pose variable, $p(k_t|k_{t-1}^{(n)}, S_t)$, are kept similar than in Section 4.2.2.

To allow jumps to highly likely regions of the state space, we can exploit the head detector in Equation 4.27 and define the proposal to be:

$$Q(X_t|X_t^{(n)}, z_t) = (1-\varepsilon)p(X_t|X_t^{(n)}) + \frac{\varepsilon}{N_d}\sum_{j=1}^{N_d} p(X_t|\check{X}_j(z_t)). \tag{4.72}$$

With this definition of the proposal, the acceptance ratio cab be expressed as:

$$a = \frac{p(z_t|X^*)\left(\sum_{j=1}^{N_s} p(X^*|X_{t-1}^j)\right) Q(X_t^{(n)}|X^*, z_t)}{p(z_t|X_t^{(n)})\left(\sum_{j=1}^{N_s} p(X_t^{(n)}|X_{t-1}^j)\right) Q(X^*|X_t^{(n)}, z_t)} \tag{4.73}$$

which still requires the computation of the costly predictive term involving the sum of the previous particles. In Section 4.6, presenting the experiments we conducted to evaluate the head pose tracking approaches we proposed, only this method using the proposal defined in Equation 4.73 will be considered, when we talk about MCMC head pose tracking.

4.4.3 Filter output

As the chain defines a pdf over the state space, we can use as output the expectation value of this pdf, obtained by standard averaging over the elements of the chain. The averaging is done in the way used to obtain the MSPF tracker output in Section 4.2.5. The only difference is that, for the MCMC method, the particles are equally weighted.

1. **initialization step:**
   for $n = 1, ..., N$: $X_0^{(n)} \sim p(X_0)$ set $t = 1$
2. **chain initialization:**
   set $X_t^(1)$
3. **chain building:**
   for $n = 2, ..., N_s$
   - sample hypothesis $X^* \sim Q(X_t)$
   - compute acceptance ratio a (see Equation 4.63)
   - sample uniformly $u_a \sim \mathcal{U}_{[0,1]}$: if $u_a \leq a$, $X_t^{(n)} = X^*$ otherwise $X_t^{(n)} = X_{t-1}^{(n)}$
4. **time step increment:**
   set $t = t + 1$ then go back to step 2

Figure 4.14: MCMC Algorithm

4.5 Head tracking then pose estimation (HTPE)

One of our interests in this thesis is to compare the *joint head tracking and pose estimation* framework to the *head tracking then pose estimation* framework. In this section, we describe a methodology to track the head using the PF framework, then estimates its pose.

4.5.1 Head tracking with a color histogram

The goal of head tracking is to estimate the pdf of the head spatial configuration (location and size) given the sequence of observations, $p(S_t|z_{1:t})$. This can be done representing the pdf with weights and samples and propagating the samples through time using IS (see Section 2.2.2). To apply the IS framework to the head location tracking problem without taking into account the head orientation, the following elements have to be defined:

- a state model,
- an orientation-independent head model,
- a dynamical model,
- an observation model,
- and a proposal density.

As in Section 4.2.1 the state space S_t modeling the spatial configuration is defined by a localization component, a scale and an eccentricity. The dynamical model $p(S_t|S_{t-1}, S_{t-2})$ and the proposal function

$q(S_t|S_{t-1}, z_t)$ are the one defined in (4.16) and (4.30) respectively [4]. In the following, we describe the head model and the observation model.

Pose-independent head model:
We define a pose-independent head model with two components. The first component is the head silhouette model described in Section 4.1.1. The second component of the model is a color histogram, E^{col}, in the HSV color space. It models the color distribution of the head given a good localization. The color histogram, E^{col}, is built from a training image and is later on used as the head template model. To obtain the histogram, for each channel, the values are binarized into 8 uniform bins. A color histogram is used because it can be considered as being pose independent. In addition, the HSV color histograms are robust to illumination variations.

The color observation $z^{col}(S)$ corresponding to a given head location S is obtained by computing the HSV color histogram of the patch image defined by the spatial configuration. The likelihood of this observation is then defined as an exponential distribution:

$$p_{col}(z^{col}|S) = \lambda_{col} \exp -\lambda_{col}\rho^2_{batt}(E^{col} - z^{col}(S)) \tag{4.74}$$

where ρ_{batt} is the Battacharya distance defined in Equation 2.9. The parameter λ_{col} controls the skewness of the likelihood, and is set to $\lambda_{col} = 10$. This parameter value is penalizing enough for state hypotheses with observations that are different from the histogram model.

Observation model:
The observation $z = (z^{sil}, z^{col})^T$ has two components: a silhouette component, z^{sil}, and an HSV color histogram component, z^{col}. Assuming the two observation components to be independent given the state, the observation likelihood is defined as:

$$p(z|S) = p_{sil}(z^{sil}|S)p_{col}(z^{col}|S) \tag{4.75}$$

where the silhouette part of the likelihood is defined by Equation 4.8 and the color likelihood is defined by Equation 4.74.

The output $\hat{S}_t$ of the head tracking at each time is taken to be average of the target distribution. It gives the location of the head at each time.

4.5.2 Pose estimation from head location

Given the head location $\hat{S}_t$, the pose can be estimated using the texture and skin color head pose models defined in Section 4.1. The estimated head pose from the location is taken to be the average of the head poses with respect to the likelihood of the texture and skin color observation, respectively defined in Equation 4.1 and 4.4, extracted from head location with respect to their exemplars. If we denote by

$$w_{k,r} = p_0(\theta_k)p(z^{tex}(\hat{S}_t, r)|k)p(z^{skin}(\hat{S}_t, r)|k) \tag{4.76}$$

[4]Only the part of the dynamical model and proposal density in Section 4.2 concerning the spatial configuration variable S_t are taken into account.

the likelihood of the texture and color observation with respect to the estimated head location, and the summation of all the likelihood terms as

$$\bar{w} = \sum_{k,r} w_{k,r}, \tag{4.77}$$

the estimated head pose $\hat{\theta}_t = (\hat{\theta}_t, \hat{\beta}_t, \hat{\gamma}_t)$ is given by:

$$\begin{aligned} \hat{\alpha}_t &= \frac{1}{\bar{w}} \sum_{k,r} w_{k,r} \alpha_k \\ \hat{\beta}_t &= \frac{1}{\bar{w}} \sum_{k,r} w_{k,r} \beta_k \\ \hat{\gamma}_t &= \frac{1}{\bar{w}} \sum_{k,r} w_{k,r} (\gamma_k + r) \end{aligned} \tag{4.78}$$

The estimated taken as an average in Equation 4.78 pose is preferred over the maximum likelihood estimate because, in practice, the average gives smoother estimates than the mode.

4.6 Head pose tracking evaluation

The previous Sections of this chapter describe the head pose tracking methods we proposed. In this section, we present our evaluation framework, then give the performances of the proposed methods.

4.6.1 Data set and protocol of evaluation

We used the part of the IHPD, recorded in a meeting room, to evaluate our tracking methods. The recordings consisted of 8 meetings where, in each of the meetings two people had their head pose annotated with a magnetic field sensor tracker. Overall, we have for evaluation 16 different people, in a meeting situation. Details about the head pose database can be found in Section 3.

The texture and skin appearence models were learned using the Pointing database. Figure 4.15 presents the 15 people appearing in this database. While providing some amount of appearence variability, it is far from being a complete representation of all appearence types that can be encountered. For instance, this database does not contain bald people, an example of which can be found on our database. The fact that there is a mismatch between the appearence of the people in the IHPD database (see Figure 3.7) and the appearances in the training data makes our task more difficult.

The tracking evaluation protocol is the following. In each of the recording of the 16 persons, we selected 1 representative minute of recording, i.e. 1500 video frames, as evaluation data. For our evaluation protocol, we used a leave one out approach. The data of one of the person was left out to be used as test set to evaluate the algorithms. The data of the other persons were used to train the few parameters of our pose dynamic model. For each test sequence, we will run our algorithm with a varying number of particles ($N_s = 100, 200, 300, 400$) to study the dependency of the algorithm to the number of samples and by the same time analyze how many particles are required for filter convergence. Figures 4.16(a)

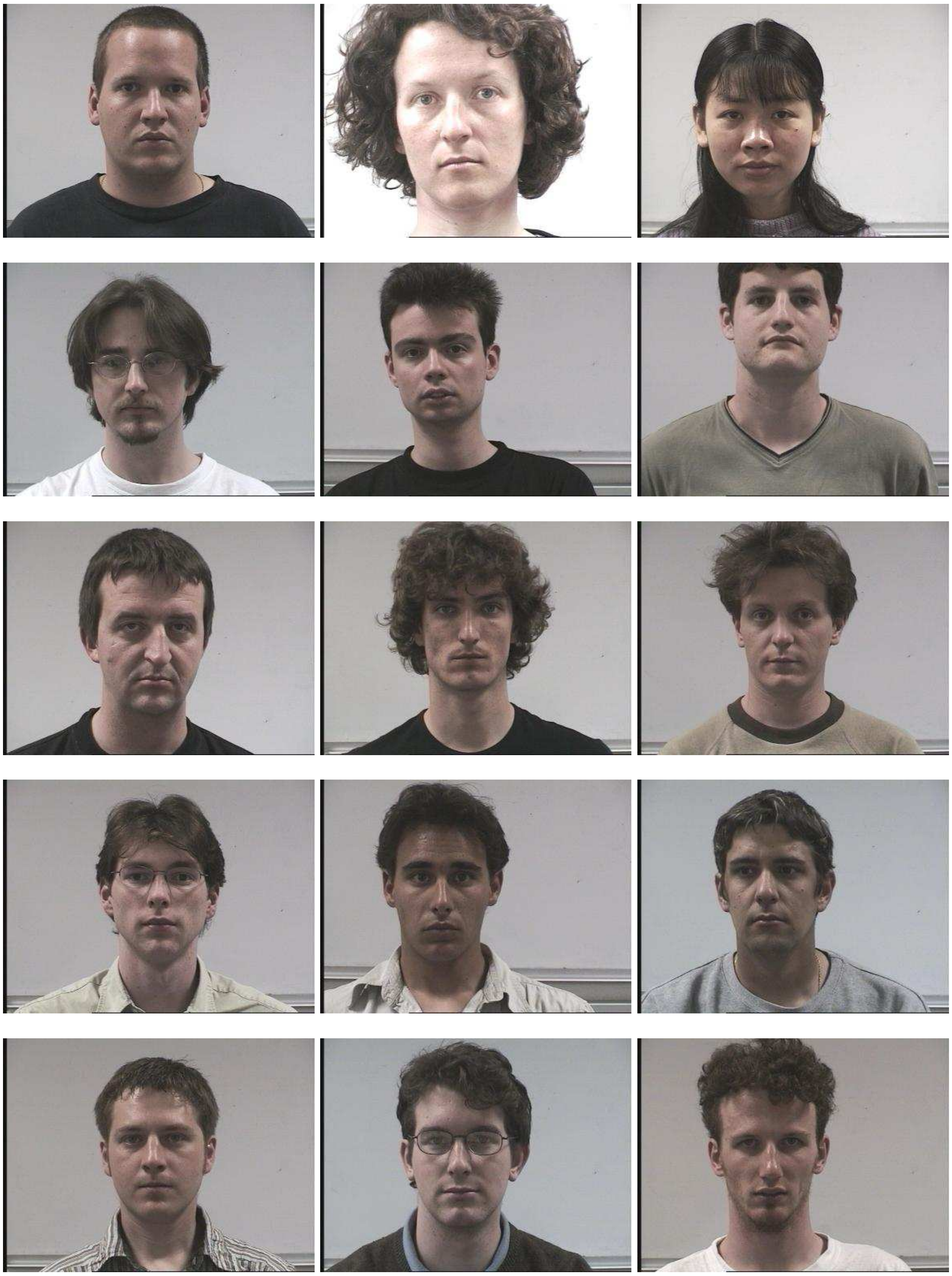
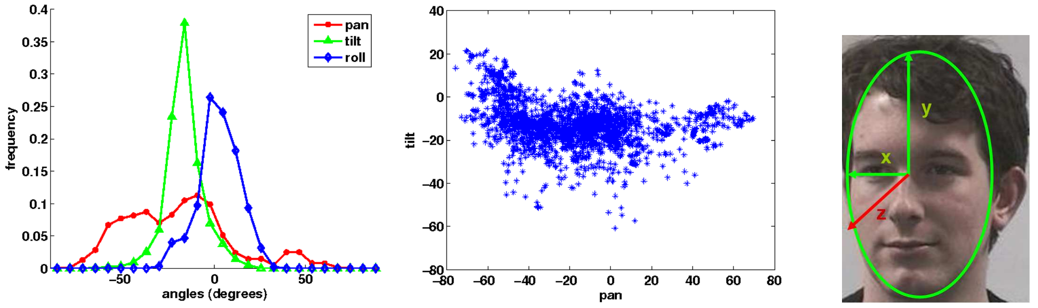


Figure 4.15: The 15 different people present in the Pointing database



(a) Head pose distribution. (b) Head pan versus head tilt scatter plot. (c) Head pointing vector.

Figure 4.16: Head pose tracking evaluation. Figures 4.16(a) and 4.16(b) shows the head pose evaluation data distribution (in the pointing representation). Figure 4.16(c) displays the head pointing vector.

and 4.16(b) show the distribution of the pan, tilt and roll values on the evaluation data. Because of the scenario used to record data, people have more frequently negative pan values, mostly corresponding to the persons looking at the projection screen located at the right of them. The majority of pan values ranges from -60 to 60 degree. Tilt values range from -60 to 15 degrees and roll value from -30 to 30 degrees.

A head pose defines a vector in the 3D space, the vector indicating where the head is pointing at as can be seen in Figure 4.16(c). It is worth noticing that in the Pointing representation, this vector depends only on the head pan and tilt angles. The angle between the 3D pointing vector defined by the head pose ground truth (GT) and the head pose estimated by the tracker can be used as the first pose estimation error measure. If, for each individual $j = 1, ..., 16$ in the evaluation data $\{v_t^{gt,j},\ t = 1, ..., 1500\}$ is the sequence of ground truth pointing vectors obtained using the magnetic field tracker, and $\{v_t^{est,j},\ t = 1, ..., 1500\}$ is the sequence of estimated pointing vector obtained using a head pose tracking method, the pointing vector average error for an individual j is computed as:

$$e_j^v = \frac{1}{1500} \sum_{t=1}^{1500} \arccos(v_t^{gt,j} v_t^{est,j}) \tag{4.79}$$

where, a each time t, $v_t^{gt,j} v_t^{est,j}$ is the scalar product of the ground truth and the estimated pointing vectors. The average pointing vector error over the whole evaluation data is taken to be the average of the individual pointing vector errors:

$$e^v = \frac{1}{16} \sum_{j=1}^{16} e_j^v. \tag{4.80}$$

This measure of error is suited for studies on the focus of attention, where the main concern is to know where the head/person is looking at. However, it gives no information about the roll estimation error. In order to have more details about the origins of the errors we will also measure the individual errors made separately on the pan, tilt and roll angles measured in the Pointing representation. We use as error measure the average of the absolute head pan, tilt, and roll estimation errors. If, for each individual $j = 1, ..., 16$ in the evaluation data $\{\alpha_t^{gt,j}, t = 1, ..., 1500\}$ is the sequence of ground truth head pan angles obtained using the magnetic field tracker, and $\{\alpha_t^{est,j}, t = 1, ..., 1500\}$ is the sequence of estimated head pan angles obtained using a head pose tracking method, the head pan average error for an individual j is computed as:

$$e_j^\alpha = \frac{1}{1500} \sum_{t=1}^{1500} |\alpha_t^{gt,j} - \alpha_t^{est,j}| \tag{4.81}$$

where, a each time t, $|\alpha_t^{gt,j} - \alpha_t^{est,j}|$ denotes the absolute value of the difference between the ground truth and the estimated head pan angles. The average head pan error over the whole evaluation data is taken to be the average of the individual head pan errors:

$$e^\alpha = \frac{1}{16} \sum_{j=1}^{16} e_j^\alpha. \tag{4.82}$$

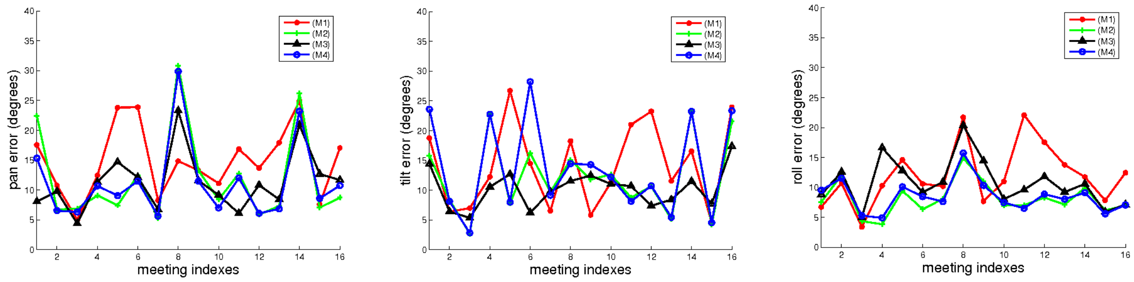
The head tilt and head roll average error are computed similarly. For each one of the four error measures, we will also compute the standard deviation, and median value of the absolute value of the errors. We used the median value because it is less sensitive to extremal values than the mean. Thus, the median value will be less biased by short time period with large pose estimation errors due to a very bad head localization. Before describing the experimental results, let us remind that all the error measures are computed in the Pointing representation.

4.6.2 Experimental results

In this chapter, we proposed four head pose tracking methods. The method described in Section 4.5, denoted M1, performs head location tracking, then estimates the pose. The three other methods perform jointly head tracking and pose estimation. The MSPF method described in Section 4.2 is denoted M2, the RBPF method described in Section 4.3 is denoted M3, and the MCMC based method sampling new states taking into account information from the observations described in Section 4.4.2 is denoted M4. We give their performance according to our evaluation framework.

Head tracking then pose estimation results

Table 4.4 shows the results obtained with the method M1 where pose estimation is performed after head localization. This Table shows that, in average, the absolute pointing vector estimation error is about 21 degrees. The average pan estimation error is about 14 degrees, the average tilt error is about 14 degrees, and the average roll errors is about 12 degrees. For all the error measures, pointing vector, pan, tilt and roll, the average estimation error is higher than the median estimation. This is due to short time failure inducing large tracking errors in the head pose estimates. For a number of particles varying from 100 to 400, the performance for the pointing vector estimation are always similar. As this hold for all angles, this means that adding more than 100 particles to the location tracker does not change its output.

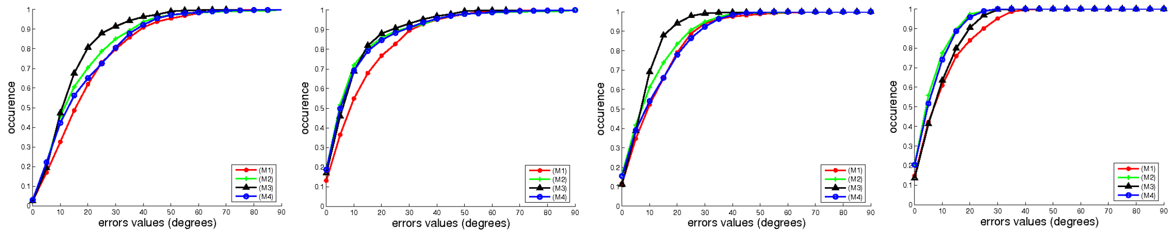


(a) Pan estimation errors for individuals.

(b) Tilt estimation errors for individuals

(c) Roll estimation errors.

Figure 4.17: Head pose tracking errors for each person using the 4 tracking methods.



(a) Pointing vector errors distribution .

(b) Pan errors distribution.

(c) Tilt errors distribution.

(d) Roll errors distribution

Figure 4.18: Cumulative distribution of the head pointing vector, the head pan, the head tilt, and the head roll estimation errors for the 4 tracking methods.

Figure 4.17 shows the average errors for each individual in the evaluation data, which shows variability of the pose estimation performances depending on the individuals. The person dependant performance variability can be explained by the fact that some people are better represented by the appearance model than others. Figure 4.19 displays head pose tracking image results for one of the person. It illustrates well the drawback of the tracking then head pose framework estimation. Despite the overall good tracking quality, in the two last images of the second row of Figure 4.19, we can notice some not good head localizations, which in general potentially imply large errors in the head pose estimation. This kind of situations motivates us to investigate jointly tracking head location and pose.

MSPF head pose tracking with the results

Table 4.5 shows the head pose tracking errors when using the method M2, which tracks the head location and pose jointly with a MSPF based on importance sampling. For this method, on average, the pointing vector estimation error is about 19 degrees, the pan estimation error is about 12 degrees, the tilt estimation error is about 13 degrees, and the roll estimation error is about 8 degrees. Comparing with the results of M1, we see that M2 is performing better than M1 according to all the performance measures, with 2

	pointing vector			pan			tilt			roll		
nparticles	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med
100	21.1	11.4	18.4	14.8	11.2	11.8	14.1	8.2	12.7	11.9	8.2	10.6
200	21.3	11.5	18.7	14.9	11.2	12	14.2	8.1	12.8	12	8.1	10.7
300	21.2	11.4	18.4	14.9	11.2	11.9	14.2	8.1	12.8	12	8.1	10.8
400	21.3	11.4	18.5	14.9	11.2	11.9	14.2	8.2	12.9	12	8.1	10.7

Table 4.4: Tracking then head pose estimation errors statistics over evaluation data for a varying number of particles.

Figure 4.19: Head pose tracking using method M1 with 200 samples. The green box represents the head localization and the red arrow represents the estimated head pointing vector.

degrees less for the pointing vector, pan and tilt errors, and 4 degrees for the head roll. The improvements in the pose estimation of (M2) over (M1) are a results of the joint search of the head location and pose.

Figure 4.18 shows the cumulative distributions of the pointing vector, the pan, the tilt and the roll estimation errors for all the tracking methods. For all angles, the cumulative errors distribution for M2 is above the cumulative distribution for the method M1. This shows that M2 outperforms M1 in term of head pose estimation.

Figure 4.17 shows the average of the pan, the tilt and the roll estimation errors, for each individual in the evaluation data taken separately, using the method (M2). Analyzing the pose estimation errors for method M2, one can notice that, among the 16 persons, 11 have a pan average error lower that 10 degrees, while for the method M1, only 3 persons have their pose estimation error lower than 10 degrees. Even for the individual taken separately, the method performing joint head location and tracking achieves better performances.

Contrarily to the M1, a slight improvement of the pose estimation performance can be noticed when increasing the number of particles with M2. The pointing vector estimation error is passing from 19 degrees when using 100 particles, to 18.5 degrees when using 400 particles. Thus, augmenting the number of particles can lead to better estimates.

	pointing vector			pan			tilt			roll		
nparticles	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med
100	19	9.7	17.5	12	9.8	9.4	13.1	7.8	12.4	8.3	5.5	7.7
200	19.2	10	18.2	12.1	10	9.3	13	7.9	12.3	8.3	5.7	7.6
300	18.4	9.7	17.1	11.8	9.8	9.4	12.2	7.7	10.9	8.1	5.6	7.4
400	18.5	9.6	17.7	11.4	9.6	8.7	12.8	7.7	11.9	8.1	5.5	8.3

Table 4.5: MSPF tracking performances
MSPF head pose tracking errors statistics for a varying number of particles

Figure 4.20: Head pose tracking using method M2 with 200 samples. The green box represents the head localization and the red arrow represents the estimated head pointing vector.

RBPF head pose tracking results

Table 4.6 shows the performance achieved by method (M3) which performs joint head location and pose estimation with the RBPF sampling approach. On average M3 achieves a pointing vector error of 15 degrees, which is significantly better than M1 (6 degrees lower). When comparing the performances of the methods M2 and M3, M3 the Rao-Blackwellized version of M2, is as expected, achieving better performances. By analyzing and comparing the pan and tilt estimation performance, we see that M3 is achieving better pointing vector estimation over M2 because its tilt estimation error is lower. For the pan estimation, M2 and M3 are performing similarly as further shown by the cumulative distributions of M2 and M3 in Figure 4.18. The cumulative distribution of the tilt errors of method M3 is above the cumulative distribution for method M2. However, the method M2 performs better than M3 in term of head roll estimation as can be seen by comparing the cumulative distribution of the head roll estimation of these two methods. Thus, the improvements due to the Rao-Blackwellization appear more on the Rao-Blackwellized variable k_t which represents the head pan an tilt.

The results in Table 4.6 also show that increasing the number of particles does not improve the performances of M3. For a number of particles ranging from a 100 to 400 particles, the filter is performing similarly. This is due to the fact that the search space is smaller for M3 than M2 due to the Rao-Blackwellization of the pose variable k_t.

	pointing vector			pan			tilt			roll		
nparticles	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med
100	15	9.8	12.6	11.2	9.9	8.6	9.1	6	8.1	11.1	7	9.8
200	15.9	8.9	13	11.3	8.9	9.3	10.2	5.9	9.5	10.8	6.4	7.6
300	15.5	8.8	12.5	11.4	9.7	9.1	10	6.3	8.9	10.1	6.8	8.1
400	15.4	9.3	12.8	11.6	8.1	9.7	10.5	6.3	9.1	10.3	7	9.6

Table 4.6: RBPF head pose tracking errors statistics for a varying number of particles.

Figure 4.21: Head pose tracking using method M3 with 200 samples. The green box represents the head localization and the red arrow represents the estimated head pointing vector.

Figure 4.17 shows that the individual person errors estimation in are general lower when using method (M3) than when using the other methods M1 or M2.

Figure 4.21 shows sample image of tracking results using the method M3. As can be seen in this Figure, method M3 is localizing better the head than the other methods (see the 2 last images of the second row). M3 is not only outperforming the other method in estimating the head pose, also it allows better head localization than the other methods.

MCMC head pose tracking results

Table 4.7 shows the head pose tracking error statistics with method (M4) which performs jointly head tracking and pose estimation based on an MCMC sampling method. In term of pose estimation performances, methods (M4) and (M2) are performing similarly. Although (M2) is based on IS and (M4) is based on MCMC. The cumulative distribution of the pose tracking errors are very similar (see Figure 4.18).

Comparison with state of the art methods

In the literature about head pose tracking we could find two head pose tracking methods that were evaluated with head pose ground truth data, in a similar fashion than our evaluation [102, 44]. We did not have

	pointing vector			pan			tilt			roll		
nparticles	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med
100	18.9	9.8	17.9	11.2	9.2	8.7	13.8	8.4	13.2	8.9	6	7.7
200	18.5	9.7	17.6	11.2	9.1	8.8	13.4	8	12.9	8.2	5.7	7.2
300	18.8	9.5	17.9	11.2	9.2	8.6	13.6	7.6	13.1	8.5	5.8	7.5
400	18.4	9.7	17.5	11.2	9.2	8.5	13.2	7.5	13	8.2	5.6	7.5

Table 4.7: MCMC head pose tracking errors statistics for a varying number of particles.

access to their data, thus, performance comparisons have to be done with cautious. For the comparison we consider only the results from our method M3.

In [44], a method was proposed for head pose tracking, to model a head in the 3D space as an ellipsoid with a set of points on its surface. Each point, indexed by i, is represented by its coordinates and a Gaussian probability distribution function $p_i(z|\theta)$ representing the probability that given a particular head pose, the point i will generate an observation z. The head models are used in a particle filter framework for head pose tracking. For their head pose tracking evaluation, 16 sequences of 11 different people were used. Head pose ground truth was determined with a polhemus tracker. Several experiments were conducted to evaluate their algorithm. Here we consider the evaluation set up similar to our's. In this setup, in turn, they left aside the data of one individual as test data, and train the appearance models with the data of the other individuals. As performance measure, the mean of the pose tracking errors (pan and tilt) are computed depending on whether the absolute value of the pan component of the head pose ground truth is lower or higher than 45 degrees. In our test data, the head poses which have a pan value between -45 and 45 degrees represent 79% of the data.

The pan and tilt errors, according to these two categories are displayed in Table 4.8. For comparison purposes this table displays also the results reported in [44] (Wu et *al*). From the results of our tracker (M3), we can conclude that pan estimation is much more reliable when the pan value is in the interval $[-45, 45]$. This shows that without common data, comparison between different algorithms must be taken with cautious. Nevertheless, according to the results, given the large difference in the performances, we can probably say that our method M3 is performing better than Wu et *al* for pan and tilt estimation in both ranges of head pose.

	abs(pan of GT)≤ 45			$45 <$ abs(pan of GT)≤ 90		
	pan	tilt	roll	pan	tilt	roll
(M3)	6.9	9.9	8.7	16.9	10.9	13.1
Wu et *al*	19.2	12.0	$\times$	33.6	16.3	$\times$

Table 4.8: mean of pan, tilt and roll errors for abs(pan of GT)≤ 45 and $45 <$ abs(pan of GT)≤ 90 (Pointing representation)

In [102], neural networks were trained to estimate the head pose in pan and tilt directions. Each network was a multilayer perceptron with one output unit (for pan or tilt), and with 20 to 60 hidden units. Output values for pan and tilt were normalized to vary between 0 and 1. Head detection and tracking was done by localizing the face using a skin color model in normalized RG color space. For the user independent evaluation set up, the head pose data of 12 individuals, among 14, were used as a training set consisting of 6080 images for training and 760 for cross-validation. The data of the 2 remaining

	Stiefelhagen et *al*	M3	M3 for abs(pan of GT) ≤ 45
pan error	9.9	11.3	6.9
tilt error	9.8	10.2	9.9
roll error	$\times$	10.8	8.8

Table 4.9: Comparison with the head pose tracking method proposed by Stiefelhagen et *al* [102].

	pointing vector			pan			tilt			roll		
nparticles	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med	mean	std dev.	med
100	18.6	11.4	15.3	13.1	11.9	9.5	10.8	6.9	9.9	10.1	7	9.2

Table 4.10: Performances of the RBPF algorithm over the whole database for 100 particles.

individuals were used as test set. The recordings wer done with an omnidirectional camera leading to poor resolution imagery. The average pan and tilt estimation errors were taken as performance measures. Table 4.9 shows the performance for head pan and tilt estimation. In comparison to our results, the performance presented in [102] are slightly better than the performances of our method (M3). However, these results have to be analyzed by keeping in mind that the results of our method (M3) is obtained from the head pose tracking of 16 different persons totalizing 16 minutes of data, while the results in [102] are only evaluated on 2 different persons corresponding to approximately 2 minutes of data. Also, our head pose models are learned on an external set up, which is not the case in [102]. Indeed, the recording set up (camera location, the illumination conditions) for the head pose modeling set up is not the same than our tracking recording set up. A further analysis of the experimental set up and the head pose distribution in [102] people's head pose lies mostly in the range abs(pan) ≤ 45 degrees. Considering only the performance in this range of pose given in Table 4.9, our method M3 performs better.

4.7 Conclusions

In this chapter, we presented 4 probabilistic methods for head pose tracking with multiple visual cues. The first method, considered as baseline, performs head location, then from features extracted around the head location estimates, the head pose. The three other methods perform head location and pose tracking jointly.

The experiments we conducted, using the head pose video database described in Chapter 3, showed that *jointly tacking the head location and pose* is more efficient than *tracking the head then estimating its pose*. Among the methods performing joint head tracking and pose estimation, the one based on RBPF sampling were outperforming the other methods, though not by a significant amount. Table 4.10 shows the results of the RBPF method over the whole database, not only the 16 minutes used for evaluation, but the whole 80 minutes of data of the meeting room recordings. The performance of the RBPF tracker are still consistent on the whole database although a slight degradation of the head pan estimation can be noticed. The RBPF based method will be used in the following chapter to generate head pose tracking estimates that will be used to study visual focus of attention for static people.

Chapter 5

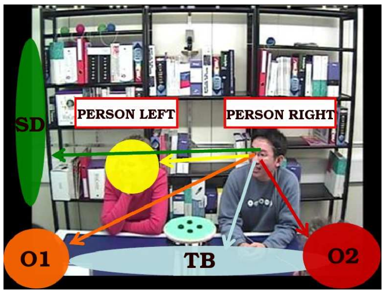
Visual focus of attention recognition in meetings

In this chapter, we investigate the recognition of the visual focus of attention of meeting participants from their head pose. In this context, the goal of this chapter is to analyze the correspondence between the head pose of people and their gaze in more natural and realistic meeting scenarios than those considered before [102, 114], and propose methods to recognize the VFOA of people from their head pose (see Figure 5.1(c), and Figure 5.7 for some results). In smart spaces such as meeting rooms, it has been claimed that head orientation can be reasonably utilized as an approximation of the gaze [102]. Gaze estimation requires high resolution close-up views, which are generally not available in practice. In this chapter, we evaluate the validity of this claim by generalizing to more complex situations similar works that have already been conducted by [102] and [114]. In the first place, unlike previous works, the scenario we consider involves people looking at slides or writing on sheet of paper on the table. As a consequence people have more potential VFOA targets in our set-up (6 instead of 3 in the cited works), leading to more ambiguities between VFOA. In the second place, due to the physical placement of the VFOA targets, the identification of the VFOA can only be done using the complete head pose representation (pan and tilt), instead of just the head pan as done previously. Thus our work addresses more natural and realistic -and hence more challenging- meeting room situations in which people do not just focus their attention on other people but also on other room targets.

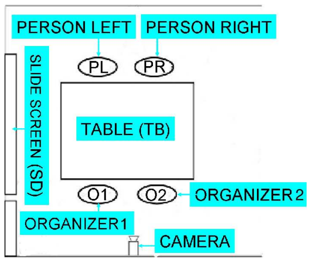
To recognize the VFOA of people from their head pose, we investigated two generative models. One that applies to single pose observation, and one, namely a Hidden Markov Model (HMM), that segments pose observation sequences into VFOA temporal segments. In both cases, the head pose observations are represented using VFOA conditional Gaussian distributions. Alternatives approaches were considered to learn the model parameters. In one approach, a machine learning point of view with training data was exploited. However, as collecting training data can become tedious, we exploited the results of studies on saccadic eye motion modeling [115, 116] and proposed a geometric approach that models the head pose of a person given his upper body pose and his effective gaze target. In this way, no training data is required to learn parameters, but knowledge of the 3D room geometry and camera calibration parameters is necessary. Finally, to account for the fact that in practice we observed that people have their own head pose preferences for looking at the same given target, we adopted an unsupervised Maximum A Posteriori (MAP) scheme to adapt the parameters obtained from either the learning or geometric model

(a) Meeting room

(b) Dataset sample Image



(c) Potential VFOA targets



(d) Meeting room geometric configuration

Figure 5.1: Recognizing the VFOA of people. 5.1(a) the meeting room, 5.1(b) a sample image of the dataset, 5.1(c) the potential VFOA targets for the left person, 5.1(d) the geometric configuration of the room.

to individual people and meetings.

To evaluate the different aspects of the VFOA modeling (model, parameters, adaptation), we have conducted comparative and thorough experiments on a the meeting room data of our head pose video database recorded in a meeting room. The database, described in Chapter 3, is comprised of 8 meetings for which both the head pose ground-truth and VFOA label ground truth are known. Due to this feature, in our experiments, we will be able to differentiate between the two main error sources in VFOA recognition: the use of head pose as proxy for gaze, and errors in the estimation of the head pose, e.g. in our case using the Rao-Blackwellized particle filter head pose tracker described in the previous Chapter.

The remainder of this chapter is organized as follows. Section 5.1 discusses the related works. Section 5.2 describes the task and the database that is used to evaluate the models we propose. Section 5.3 reminds our probabilistic method for joint head tracking and pose estimation, and discussed its performance by comparison with the pose ground truth. Section 5.4 describes the considered models for recognizing the VFOA from head pose. Section 5.5 gives the unsupervised MAP framework used to adapt our VFOA models to unseen data. Section 5.6 presents our evaluation setup. We give experimental results in Section 5.7 and conclusions in Section 7.

5.1 Related work

In this work, we investigate the VFOA recognition from head pose in the meeting contexts. Thus, we had to analyze the related works along the following lines: technologies that can be used in general for VFOA estimation, the estimation of head pose from vision sensors, and the actual recognition of the VFOA from head pose. In Chapter 2, the first two points had already been adressed. In this Section we review only works based on computer vision techniques. [81] gives a good overview about computer vision based gaze tracking systems. Computer vision techniques are required in situations where the invasiveness of sensor based gaze estimation methodologies becomes sensitive. For instance, in automatic drivers (loss of) attention detection applications [117], and more generally in the human computer interface (HCI) domain [118], eye-appearance based gaze tracking systems can be used. These systems generally rely on the availability of high resolution images to locate the eye balls and reconstruct the gaze from the estimated iris location within the eye-white area. In [117], motion and skin color distributions are used to track facial features and eye ball characteristics to estimate a driver's gaze direction. In [118], a similar approach is used to estimate the gaze direction of a worker siting in front of his computer in an office environment. Although less invasive than wearable sensor approaches, such computer vision technics are still relatively constraining and usually restrict the mobility of the subject, since the need for high resolution close-up face images requires cameras with narrow field-of-views.

An alternative to the previously described approaches is to use the head pose as a surrogate for gaze. Using the head pose as a surrogate for gaze is possible even in some extreme cases such as showed in [119]. In [119], the gaze was estimated on tentatively very low resolution images. The head height lays between 20 and 40 pixels. In the meeting contexts, [102] used an HMM to model the sequence of head pose observations. The GMM model were learned on the test data after initialization from the output of a K-means clustering of the pose values. This approach was possible due to the employed meeting physical set-up (four participants evenly spaced around a round table) and the limitation of the VFOA targets of a person to the other participants, which allowed to assimilate the head pose with the head pan value only, and drastically restrained the possibility of ambiguities. More recently, [114] used a dynamic Bayesian network to jointly recognize the VFOA of people as well as different conversational models in a 4 persons conversation, based on head pan and utterance status observations. However, the pan information was obtained using a magnetic field head orientation sensor, and the utterance status was annotated manually. In another interesting work [120], the head pose extracted from an overhead camera tracking retro-reflective markers mounted on headsets was exploited to look for shared mutual visual attention situation. This information is then exploited to derive the social geometry of co-workers within an office, and infer their availability status for communication.

5.2 Database and task

In this section, we describe the task and the data that is used to evaluate both our pose estimation algorithm and VFOA recognition algorithm.

5.2.1 The task and VFOA set

In this work, our goal is to evaluate how well we can infer the VFOA state of a person using head pose in real meeting situations. There are two important issues. The first issue is that by definition, the

VFOA is given by the eye gaze. In practice, however, psycho-visual studies have shown that people use other cues -e.g. head and body posture, speaking status- to recognize the VFOA state of another person [1]. Thus, one general objective is to see how well one can still recognize the VFOA of people from these other cues, in the absence of direct gazing measurements, a situation likely to occur in many applications of interest. The second issue is what should be the definition of a person's VFOA state ? At first thoughts, one can consider that any gaze direction values could correspond to a potential VFOA. However, studies about the VFOA in natural conditions [121] have shown that humans tend to look at targets in their environment that are either relevant to the task they are solving or of immediate interest to them. Additionally, one interprets another person's gaze not as continuous spatial locations of the 3D space, but as gaze towards objects that has been identified as potential targets. This process is often called the shared-attentional mechanism [122, 1], and suggests that in general VFOA states correspond to a finite set of targets of interests.

Taking into account the above elements, we define our task more precisely our task as the following: given the head orientation of a person, how to infer his VFOA state. In the context of our meeting set-up and database (see below), the set of potential VFOA targets of interest, denoted $\mathcal{F}$, has been defined as: the other participants to the meeting, the slide-screen, the table, and an additional label (unfocused) when none of the previous could apply. As a result, for the 'person left' in Figure 5.1(c), we have: $\mathcal{F} = \{PR, O2, O1, SS, TB, U\}$ where PR stands for person right, $O1$ and $O2$ for organizer 1 and 2, SS for slide screen, TB for table and U for unfocus. For the person right, we have $\mathcal{F} = \{PL, O2, O1, SS, TB, U\}$, where PL stands for person left.

5.2.2 The database

Our experiments rely on the meeting room recordings of the IDIAP Head Pose Database (IHPD) described in Section 3. In the database, head pose ground truth are obtained using a magnetic field location and orientation tracker. Besides, for the same data, people's discrete VFOA was hand annotated on the basis of their gaze direction. This allows us to evaluate the impact of using the estimated vs the true head pose as input to the VFOA recognition algorithms.

Content description:
the database comprises 8 meetings involving 4 people, recorded in a smart meeting room (cf Figure 5.1(a)). The meeting durations ranged from 7 to 14 minutes, which was long enough to realistically represent general meeting scenario. With short recordings (less than 2-3 minutes), participants tend to overact with the effect of using their head to a larger extent to focus on other people/objects. In longer situations like here, the attention of participants sometimes drops and people may listen without focusing to the speaker. Note however that the small group size ensures enough engagement of participants in the meeting, in contrast to meeting with larger groups. With respect to the scenario, it consisted in writing down one's name on a sheet of paper, and then discussing statements displayed on the projection screen. There were restrictions neither on head motions, nor on head poses.

VFOA annotation:
using the predefined VFOA discrete set of targets $\mathcal{F}$, the VFOA of each person (PL and PR) was manually annotated by a single annotator using a multimedia interface. The annotator had access to all data streams, including the central camera view (Figure 5.1(a)) and the audio. Specific guidance for annotation were defined by [123]. Quality of annotation was evaluated indirectly, on 15 minutes of similar data

(a) Head Patch.

(b) Face patch.

Figure 5.2: Head patch and and face patch.

(same room, same VFOA set). Inter-annotator annotation showed good agreement, with a majority of kappa values higher than 0.8.

5.3 Head pose tracking

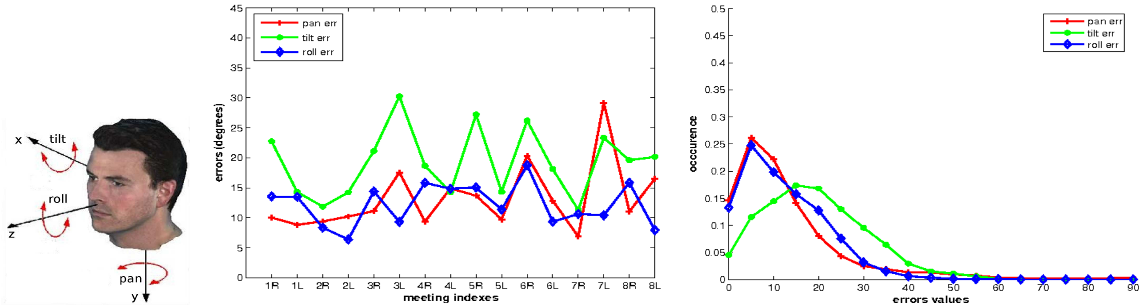
The head poses are obtained by two means: first, from the magnetic sensor readings (cf Chapter 3). This virtually noise-free version is called the ground truth (GT). Secondly, by applying a head pose tracker on the video stream. In this Section, we summarize the computer vision probabilistic tracker that we employed and which is similar to the tracking system described in more detail in Section 4. The differences are going to be pointed out in the following. Then, the pose estimation results provided by this tracker are compared with the ground truth and analyzed in detail, allowing ultimately to have a better insight in the VFOA recognition results.

5.3.1 Probabilistic method for head pose tracking

The Bayesian solution of tracking using particle filter (PF) was described in details in Section 4. Five elements are important in defining a PF: i) head pose appearance models ii) a state model defining the object we are interested in; iii) a dynamical model $p(X_t|X_{t-1})$ governing the temporal evolution of the state; iv) a likelihood model measuring the adequacy of data given the proposed configuration of the tracked object; and v) a sampling mechanism which have to propose new configurations in high likelihood regions of the state space. These elements along with our model are described in the next paragraphs.

Head pose appearance models:
It is important to note that the pose dependent appearance models were not learned using the same people or head images gathered in the same meeting room environment. We used the Pointing database [48]. Contrarily to Section 4.1 where when learning appearance models we used whole head patches, in this work , we trained the appearance models on face patches instead of head patches, as illustrated by Figure 5.2.

State space:
The state space $X = (S, \gamma, k)$ is defined as in Section 4.2.1.



(a) Head Pose definition

(b) Single person average errors, R for person right and L for person left person.

(c) Pose estimation errors distribution

Figure 5.3: Figure shows 5.3(a) head pose Euler rotation angles. Note that the z axis indicates the head pointing direction. Figures 5.3(b) and 5.3(c) display the pan, tilt and roll tracking errors with b) average errors for each person and c) distribution of tracking errors over the whole dataset.

Dynamical model:
It governs the temporal evolution of the state, and is defined as

$$p(X_t|X_{1:t-1}) = p(\gamma_t|\gamma_{t-1}, k_t)p(k_t|k_{t-1}, S_t)p(S_t|S_{t-1}, S_{t-2}). \tag{5.1}$$

Contrarily to the dynamical model defined in Section 4.2.2, the dynamical model does not comprise a the prior on the head pose, $p_0(\theta_{k_t})$, centered on the frontal head pose.

Observation model:
This model $p(Y|X)$ measures the likelihood of the observation for a given state value. The observations $Y = (Y^{tex}, Y^{skin})$ are composed by the texture and skin color observations described in Section 4. The observation model does not comprise the binary features from background subtraction. Assuming that, given the state the observation are independent, the observation likelihood is modeled as:

$$p(Y|X = (S, \gamma, k)) = p_{tex}(Y^{tex}(S, \gamma)|k)p_{skin}(Y^{skin}(S, \gamma)|k) \tag{5.2}$$

using only the texture and skin color likelihood defined in Section 4.1.

Sampling method:
In this work, we use Rao-Blackwellization sampling method described in Section 4.3.

5.3.2 Head pose tracking evaluation

Protocol:
We used a two-fold evaluation protocol, where for each fold, we used the data of 1 person in our IHPD database (see Sec.5.2.2) as training set to learn the pose dynamic model and, the data of the remaining person was used as test set.

condition	right persons		left persons		pan near frontal ($\|\alpha\| < 45°$)		pan near profile ($\|\alpha\| > 45°$)		tilt near frontal ($\|\beta\| < 30°$)		tilt far from frontal ($\|\beta\| > 30°$)	
stat	mean	med	mean	med	mean	med	mean	med	mean	med	mean	med
pan	11.4	8.9	14.9	11.3	11.6	9.5	16.9	14.7	12.7	10	18.6	15.9
tilt	19.8	19.4	18.6	17.1	19.7	18.9	17.5	17.5	19	18.8	22.1	21.4
roll	14	13.2	10.3	8.7	10.1	8.8	18.3	18.1	11.7	10.8	18.1	16.8

Table 5.1: Pan, tilt, and roll error statistics for different conditions (person left/right) and configuration of the true head pose.

Performance measures:
the three error measures, described in Section 4.6, are used. They are the average errors in pan, tilt and roll angles, i.e. the average over time and meeting of the absolute difference between the pan, tilt and roll of the ground truth (GT) and the tracker estimation. We also report the error median value, which should be less affected by very large errors due to erroneous tracking.

head pose tracking results:
The statistics of the errors are shown in Table 5.1. Overall, given the small head size, and the fact that the appearance training set is composed of faces recorded in an external set up (different people, different viewing and illumination conditions), the results are quite good, with a majority of head pan errors smaller than 12 degrees (see Figure 5.3). In addition, Table 5.1 shows that the tracking performances are better for the person sitting to the right.
Table 5.1 further details the errors depending on whether the true pose is near frontal or not. We can observe that, in near frontal position ($|\alpha| \leq 45°$ or $|\beta| \leq 30°$), the head pose tracking estimates are more accurate, in particular for the pan and roll value. This can be understood since near profile poses, a pan variation introduces much less appearance changes than the same pan variation near a frontal view. Similarly, for high tilt values, the face-image distortion introduced by the perspective shortening rotation affects the quality of the observations.

5.4 Visual focus of attention modeling

In this Section, we first describe the models used to recognize the VFOA from the head pose measurements, then the two alternatives we adopted to learn the model parameters.

5.4.1 Modeling VFOA with a Gaussian mixture model (GMM)

Let $X_t \in \mathcal{F}$ and z_t respectively denote the VFOA state and the head pointing direction of a person at a given time instant t. The head pointing direction is defined by the head pan and tilt angles, i.e. $z_t = (\alpha_t, \beta_t)$, since the head roll has in principle no effect on the head direction (see Figure 5.3(a)). Estimating the VFOA can be posed in a probabilistic framework as finding the VFOA state maximizing the a posteriori probability:

$$\hat{X}_t = \arg\max_{X_t \in \mathcal{F}} p(X_t|z_t) \text{ with } p(X_t|z_t) = \frac{p(z_t|X_t)p(X_t)}{p(z_t)} \propto p(z_t|X_t)p(X_t) \tag{5.3}$$

For each possible VFOA $f_i \in \mathcal{F}$ which is not *unfocus*, $p(z_t|X_t = f_i)$, which expresses the likelihood of the pose observations for the VFOA state f_i, is modeled as a Gaussian distribution $\mathcal{N}(z_t; \mu_i, \Sigma_i)$ with mean μ_i and full covariance matrix Σ_i. Besides, $p(z_t|X_t = unfocus) = u$ is modeled as a uniform distribution with $u = \frac{1}{180 \times 180}$ as the head pan and tilt angle can vary from -90°to 90°. In Equation 5.3, $p(X_t = f_i) = \pi_i$ denotes the prior information we have on the VFOA target f_i. Thus, in this modeling, the pose distribution is represented as a Gaussian Mixture Model (plus one uniform mixture), with the mixture index denoting the focus target:

$$p(z_t|\lambda_G) = \sum_{X_t} p(z_t, X_t|\lambda_G) = \sum_{X_t} p(z_t|X_t, \lambda_G)p(X_t|\lambda_G) = \sum_{i=1}^{K-1} \pi_i \mathcal{N}(z_t; \mu_i, \Sigma_i) + \pi_K u \quad (5.4)$$

where $\lambda_G = \{\mu = (\mu_i)_{i=1:K-1}, \Sigma = (\Sigma_i)_{i=1:K-1}, \pi = (\pi_i)_{i=1:K}\}$ represents the parameter set of the GMM model. Figure 5.11 illustrates how the pan-tilt space is split into different VFOA regions when applying the decision rule of Equation 5.3 with the GMM modeling.

5.4.2 Modeling VFOA with a hidden Markov model (HMM)

The GMM approach does not account for the temporal dependencies between the VFOA events. To introduce such dependencies, we consider the Hidden Markov Model. Denoting by $X_{0:T}$ the VFOA sequence, and by $z_{1:T}$ the observation sequence, the joint posterior probability density function of states and observations can be written:

$$p(X_{0:T}, z_{1:T}) = p(X_0) \prod_{t=1}^{T} p(z_t|X_t)p(X_t|X_{t-1}) \quad (5.5)$$

In this equation, the emission probabilities $p(z_t|X_t = f_i)$ are modeled as in the previous case (i.e. Gaussian distributions for regular VFOA, uniform distribution for the *unfocus* VFOA). However, in the HMM modeling, the static prior distribution on the VFOA targets is replaced by a discrete transition matrix $A = (a_{i,j})$, defined by $a_{i,j} = p(X_t = f_j|X_{t-1} = f_i)$, which models the probability of passing from a focus f_i to a focus f_j. Thus, the set of parameters of the HMM model is $\lambda_H = \{\mu, \Sigma, A = (a_{i,j})_{i,j=1:K}\}$. With this model, given the observations sequence, the VFOA recognition is done by estimating the optimal sequence of VFOA which maximizes $p(X_{0:T}|z_{1:T})$. This optimization is efficiently conducted using the Viterbi algorithm [113].

5.4.3 Parameter learning using training data

Since in many meeting settings, people are most of the time static and seated at the same physical positions, setting the model parameters can be done by using a traditional machine learning approach. Thus, given training data with VFOA annotation and head pose measurements, we can readily estimate all the parameters of the GMM or HMM models. Parameters learnt with this training approach will be denoted with a l superscript. Note that μ_i^l and Σ_i^l are learnt by first computing the VFOA means and covariances per meeting and then averaging the results on the meetings belonging to the training set.

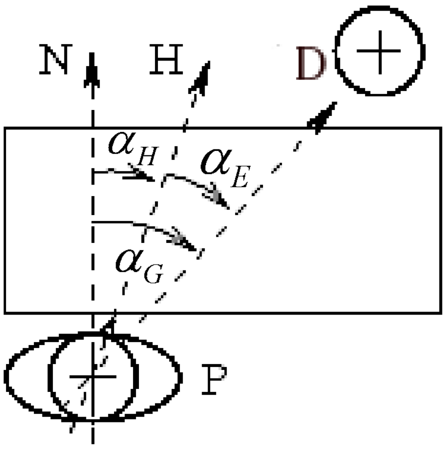


Figure 5.4: Model of gazing and head orientation.

Prior distribution and transition matrix:
While the estimation of the Gaussian parameters using training data seems appropriate, learning the VFOA prior distribution π or transition matrix A using annotated data can be problematic. If the training data exhibit specific meeting structure, as it is the case in our database where the main and secondary organizers always occupy the same seats, the learned prior will have a beneficial effect on the recognition performances for similar unseen meetings. However, at the same time, this learned prior can considerably limit the generalization to other data sets, since by simply having participants with different roles exchanging their seating positions, we can obtain meeting sessions with very different prior distributions. Thus, we investigated alternatives that avoid favoring any meeting structures. In the GMM case, this was done by considering a uniform distribution (denoted π^u) over the prior π. In the HMM case, the transition matrix was designed to exhibit a uniform stationary distribution. Self-transitions defining the probability of keeping the same focus were favored and transitions to other focus were distributed uniformly according to: $a_{i,i} = \epsilon < 1$, and $a_{i,j} = \frac{1-\epsilon}{K-1}$ for $i \neq j$. We will denote by A^u the transition matrix built in this way.

5.4.4 Parameter learning using a geometric model

The training approach to parameter learning is straightforward to apply when annotated data is available. However, annotating the VFOA of people in video recording is tedious and time consuming, as training data needs to be gathered and annotated for each (location, VFOA target) couple, the number of which can grow quickly, especially if some moving people are involved. Thus, to avoid the need for annotation, one can seek for an alternative approach that exploits the geometric nature of the problem. The parameters set with the geometric approach described below will be denoted with a superscript g (e.g. μ_i^g).

Assuming a calibrated camera w.r.t. to the room, given a head location and a VFOA target location, it is possible to derive the Euler angles (w.r.t. the camera) for which the head would be fully oriented toward the VFOA target. However, as gazing at a target is usually accomplished by rotating both the eyes ('eye-in-head' rotation) and the head in the same direction, in reality, the head is only partially oriented towards the gaze. In neurophysiology and cognitive sciences, researchers working on the modeling of the dynamics of the head/eye motions involved in saccadic gaze shifts have found that the relative contribution of the head and eyes towards a given gaze shift follows simple rules [115, 121]. While the experimental framework employed in these papers do not completely match the meeting room scenario, we have exploited these findings to propose a model for predicting a person's head pose given her gaze target.

The proposed geometric model is presented in Figure 5.4. Given a person P whose rest (or reference) head pose corresponds to looking straight ahead in the N direction, and given that she is gazing towards D, the head points in direction H according to:

$$\alpha_H = \kappa_\alpha \, \alpha_G \qquad \text{if } |\alpha_G| < \xi_\alpha, \qquad 0 \text{ otherwise} \tag{5.6}$$

where α_G and α_H denotes respectively the pan angle to look at the gaze target and the actual pan angle of the head pose, both w.r.t. the reference direction N. The parameters of this model, κ_α and ξ_α, are constants independent of the VFOA gaze target, but usually depend on individuals [115]. While there is a consensus among researchers about the linearity aspect of the relation between the gaze direction and the head pose direction described by Equ. 5.6, some researchers reported observing head movements for all VFOA gaze shift amplitude (i.e. ξ_α=0), while others do not. In this paper, we will assume $\xi_\alpha = 0$. Besides, Equ. 5.6 is only valid if the contribution of the eyes to the gaze shift (given by $\alpha_E = \alpha_G - \alpha_H$) do not exceed a threshold, usually taken at $\sim 35^o$. Finally, in [115], it is shown that the tilt angle β follows a similar linearity rule. However, in this case, the contribution of the head to the gaze shift is usually lower than for the pan case. Typical values range from 0.2 to 0.5 for κ_β, and 0.5 to 0.8 for κ_α.

In the experiments, we will test the use of this geometric model to predict the mean angles μ in the VFOA modeling. As for the rest reference direction N (Fig 5.4), we will assume that for the people seated at the two tested positions, it corresponds to looking straight in front of them. Thus, for person left (resp. right), N consists of looking at organizer 1 (resp. 2), as shown in Figure 5.1(d). The covariances Σ of the Gaussian distributions were set according to the object size and the VFOA location (near or not frontal poses, cf Table 5.1). Finally, the parameter setting of the prior follows the same considerations than in the previous subsection.

5.5 VFOA models adaptation

In the previous Section, we proposed two models (GMM and HMM) to recognize the VFOA of people from their head pose, along with two approaches to learn their VFOA target dependent parameters/ In all cases, the resulting models are generic and can be applied indifferently to any new person seated at the location related to a learned model.

In practice, however, we observed that people have personal ways of looking to targets. For example, some people use more their eye-in-head rotation capabilities and turn less their head towards the focused target than others (e.g. right person in Figure 5.5(a)). In addition, our head pose tracking system is sensitive to the appearance of people, and can introduce a systematic bias in the estimated head pose for a given person, especially in the estimated head tilt. As a consequence, the parameters of the generic models might not be the best for a given person. As a remedy we propose to exploit the Maximum A

(a) (b) (c)

Figure 5.5: Several people behaviors. Images 5.5(a) and 5.5(b): in both images, the person on the right looks at the $O1$ target. In 5.5(b), however, the head is used more importantly than in 5.5(a). - Image 5.5(c): person right turns himself and his shoulder towards $O1$ instead of towards the opposite side of the table (i.e. towards $O2$). The reference direction N (cf Fig. 5.4) should be changed accordingly.

Posteriori (MAP) estimation principle to adapt, in an unsupervised fashion, the generic VFOA models to the data of each new meeting, and thus produce models adapted to individual person's characteristics.

5.5.1 VFOA maximum a posteriori (MAP) adaptation

The MAP adaptation principle is the following. Let $z = z_1..., z_T$ denotes a set of T samples (i.i.d or drawn from a Markov chain), and $\lambda \in \Lambda$ the parameter vector to be estimated from these sample data. The MAP estimate $\hat{\lambda}$ of the parameters is then defined as:

$$\hat{\lambda} = \arg\max_{\lambda \in \Lambda} p(\lambda|z) = \arg\max_{\lambda \in \Lambda} p(z|\lambda)p(\lambda) \tag{5.7}$$

where $p(z|\lambda)$ is the data likelihood and $p(\lambda)$ is the prior on the parameters. The choice of the prior distribution is crucial for the MAP estimation. In [124] it is shown that for the GMM and HMM models, by selecting the prior pdf on λ as the product of appropriate conjugate distributions of the data likelihood[1], then the MAP estimation can also be solved using the Expectation-Maximization (EM) algorithm, as detailed in the next two Subsections.

5.5.2 GMM MAP adaptation

In the GMM VFOA model case, the data likelihood is $p(z|\lambda_G) = \prod_{t=1}^{T} p(z_t|\lambda_G)$, where $p(z_t|\lambda_G)$ is the mixture model given in Equ. 5.4, and λ_G are the parameters to learn (the multinomial prior distribution on the VFOA indices π and the Gaussian parameters μ and Σ). Note that having a uniform distribution as one mixture does not modify the GMM MAP adaptation framework.

Priors distribution on parameters:
For this model, there does not exist a joint conjugate prior density for the parameters λ_G. However, it is

[1] A prior distribution $g(\lambda)$ is the conjugate distribution of a likelihood function $f(z|\lambda)$ if the posterior $f(z|\lambda)g(\lambda)$ belongs to the same distribution family than g.

possible to express the prior probability as a product of individual conjugate priors [124]. Accordingly, the conjugate prior of the multinomial mixture weights is the Dirichlet distribution $\mathcal{D}(\nu w_1, \ldots, \nu w_K)$ whose pdf is given by:

$$p^{\mathcal{D}}_{\nu w_1,\ldots,\nu w_K}(\pi_1, \ldots, \pi_K) \propto \prod_{k=1}^{K} \pi_i^{\nu w_i - 1} \tag{5.8}$$

Additionally, the conjugate prior for the Gaussian mean and covariance matrix inverse of a given mixture is the Normal-Wishart distribution, $\mathcal{W}(\tau, m_i, d, V_i)$ $(i = 1, ..., K-1)$, with pdf

$$p_i^{\mathcal{W}}(\mu_i, \Sigma_i^{-1}) \propto |\Sigma_i^{-1}|^{\frac{d-p}{2}} \exp\Big(-\frac{\tau}{2}(\mu_i - m_i)'\Sigma_i^{-1}(\mu_i - m_i)\Big) \times \exp(-\frac{1}{2}tr(V_i\Sigma_i^{-1})),\ d > p \tag{5.9}$$

where $(\mu_i - m_i)'$ denotes the transpose of $(\mu_i - m_i)$, and p denotes the samples' dimension (in our case, $p = 2$). Thus the prior distribution on the set of all the parameters is defined as

$$p(\lambda_G) = p^{\mathcal{D}}_{\nu w_1,\ldots,\nu w_K}(\pi_1, \ldots, \pi_K) \prod_{i=1}^{K-1} p_i^{\mathcal{W}}(\mu_i, \Sigma_i^{-1}). \tag{5.10}$$

EM MAP estimate: The MAP estimate $\hat{\lambda_G}$ of the distribution $p(z|\lambda_G)p(\lambda_G)$ can be computed using the EM algorithm by recursively applying the following computations (see Fig. 5.6) [124]:

$$\begin{aligned} c_{it} &= \frac{\hat{\pi}_i p(z_t|\hat{\mu_i}, \hat{\Sigma}_i)}{\sum_{j=1}^{K} \hat{\pi}_j p(z_t|\hat{\mu_j}, \hat{\Sigma}_j)} \\ c_i &= \sum_{t=1}^{T} c_{it} \qquad (5.11) \\ \bar{z}_i &= \frac{1}{c_i}\sum_{t=1}^{T} c_{it} z_t \\ S_i &= \frac{1}{c_i}\sum_{t=1}^{T} c_{it}(z_t - \bar{z}_i)(z_t - \bar{z}_i)' \qquad (5.12) \end{aligned}$$

where $\hat{\lambda_G} = (\hat{\pi}, (\hat{\mu}, \hat{\Sigma}))$ denotes the current parameter fit. Given these coefficients, the M step re-estimation formulas are given by:

$$\begin{aligned} \hat{\pi}_i &= \frac{\nu w_i - 1 + c_i}{\nu - K + T}, \\ \hat{\mu}_i &= \frac{\tau m_i + c_i \bar{z}_i}{\tau + c_i} \\ \hat{\Sigma}_i &= \frac{V_i + c_i S_i + \frac{c_i \tau}{c_i + \tau}(m_i - \bar{z}_i)(m_i - \bar{z}_i)'}{d - p + c_i} \end{aligned} \tag{5.13}$$

- Initialization of $\hat{\lambda_G}$: $\hat{\pi}_i = w_i,\ \hat{\mu}_i = m_i,\ \hat{\Sigma}_i = V_i/(d-p)$
- EM: repeat until convergence:
  1. Expectation: compute c_{it}, $\bar{z}_i$ and S_i (Eq. 5.11 and 5.12) using the current parameter set $\hat{\lambda_G}$
  2. Maximization: update parameter set $\hat{\lambda_G}$ using the re-estimations formulas (Equations 5.13)

Figure 5.6: GMM adaptation algorithm iterations

For the uniform component ($i = K$), the appropriate uniform distribution is used in c_{it} (i.e $p(z_t|\hat{\mu_K}, \hat{\Sigma}_K)$ is indeed a uniform density), and, accordingly, only the prior weight π_K needs to be updated. The setting of the hyper-parameters of the prior distribution $p(\lambda_G)$ in Eq. 5.10, which is discussed at the end of this Section, is important as the adaptation is unsupervised, and thus only the prior prevents the adaptation process to deviate from meaningful VFOA distributions.

5.5.3 VFOA MAP HMM adaptation

The VFOA HMM can also be adapted in an unsupervised way to new test data using the MAP framework [124]. The parameters to adapt in this case are the transition matrix and the emission probabilities parameters $\lambda_H = \{A, (\mu, \Sigma)\}$[2].
The adaptation of the HMM parameters leads to a procedure similar to the GMM adaptation case. Indeed, the prior on the Gaussian parameters follows the same Normal-Wishart density (Eq. 5.9), and the Dirichlet prior on the static VFOA prior π is replaced by a Dirichlet prior on each row $p(.|X = f_i) = a_{i,\cdot}$ of the transition matrix. Accordingly, the full prior is:

$$p(\lambda_H) \propto \prod_{i=1}^{K} p^{\mathcal{D}}_{\nu b_{i,1},\ldots,\nu b_{i,K}}(a_{i,1}, \ldots, a_{i,K}) \prod_{i=1}^{K-1} p_i^{\mathcal{W}}(\mu_i, \Sigma_i^{-1}) \tag{5.14}$$

Then the EM algorithm to compute the MAP estimate can be conducted as follows. For a sequence of observations, $z = (z_1, ..., z_T)$, the hidden states are now composed of a corresponding state sequence $X_1, .., X_T$, which allows to compute the joint state-observation density (cf Eq. 5.5). Thus, in the E step, one need to compute $\xi_{i,j,t} = p(X_{t-1} = f_i, X_t = f_j|z, \hat{\lambda_H})$ and $c_{i,t} = p(X_t = f_i|z, \hat{\lambda_H})$, which respectively denote the joint probability of being in the state f_i and f_j at time $t-1$ and t, and the probability of being in state f_i at time t, given the current model $\hat{\lambda_H}$ and the observed sequence z. These values can be obtained using the Baum-Welch forward-backward algorithm [113]. Given these values, the re-estimation formulas are the same than in Eq. 5.13 for the mean and covariance matrices and as

[2] For convenience, we assumed that the initial state distribution followed a uniform distribution.

follows for the transition matrix parameters:

$$\hat{a}_{i,j} = \frac{\nu b_{i,j} - 1 + \sum_{t=1}^{T-1} \xi_{i,j,t}}{\nu - K + \sum_{j=1}^{K} \sum_{t=1}^{T-1} \xi_{i,j,t}}. \tag{5.15}$$

The discussion about how to select the hyper-parameters is conducted in the following.

5.5.4 Choice of prior distribution parameters

In this Section we discuss the impact of the hyper-parameter setting on the MAP estimates, through the analysis of the re-estimation formula (Eq. 5.13). Before going into details, recall that T denotes the size of the data set available for adaptation, and K is the number of VFOA targets.

Parameter values for the Dirichlet distribution:
The Dirichlet distribution is defined by two kinds of parameters: a scale factor ν and the prior values on the mixture weights w_i (with $\sum_i w_i = 1$). The scale factor ν controls the balance between the mixture prior distribution w and the data. If ν is small (resp. large) with respect to $T-K$, the adaptation is dominated by the data (resp. by the prior, i.e. almost no adaptation occurs). When $\nu = T-K$, data and prior contribute equally to the adaptation process. In the experiments, the hyper-parameter ν will be selected through cross-validation among the values in $C^{\nu} = \{\nu_1 = T-K,\ \nu_2 = 2(T-K)\ , \nu_3 = 3(T-K)\}$. The priors weights w_i, on the other hand, are defined according to the prior knowledge we have on the VFOA targets distribution. More likely VFOA targets such as the person who speak the most or the slide screen should be given a higher weight. When we want to enforce no knowledge about the VFOA targets distribution, the w_i can be set uniformly equal to $\frac{1}{K}$.

Parameter values for the Normal-Wishart distribution:
This distribution defines the prior on the mean μ_i and covariance Σ_i of one Gaussian. The adaptation of the mean is essentially controlled by two parameters (see Eq. 5.13): the prior value for the mean, m_i, which will be set to the value computed using either the learning ($m_i = \mu_i^l$) or the geometric approach ($m_i = \mu_i^g$) and a scalar τ, which linearly controls the contribution of the prior m_i to the estimated mean. As the average value for c_i, is $\frac{T}{K}$, in the experiments, we will select τ though cross-validation among the values in $C^{\tau} = \{\tau_1 = \frac{T}{2K},\ \tau_2 = \frac{T}{K},\ \tau_3 = \frac{2T}{K},\ \tau_4 = \frac{5T}{K}\}$. Thus, with the first value τ_1, the mean adaptation is on average dominated by data. With τ_2, the adaptation is balanced between data and prior, and with the two last values, adaptation is dominated by the priors on the means.
The prior on the covariance is more difficult to set. It is defined by the Wishart distribution parameters, namely the prior covariance matrix V_i and the number of degree of freedom d. From Eq. 5.13, we see that the data covariance and the deviation of the data mean from the mean prior also influence the MAP covariance estimate. As prior Wishart covariance, we will take $V_i = (d-p)\tilde{V}_i$, where $\tilde{V}_i$ is respectively either Σ_i^l or Σ_i^g, the covariance of target f_i estimated using either labeled training data (Subsection 5.4.3) or the geometrical VFOA considerations (Subsection 5.4.4). The weighting $(d-p)$ is important, as it allows V_i to be of the same order of magnitude than the data variance $c_i S_i$, as far as c_i and $(d-p)$ are of similar order of magnitude as well. In the experiments, we will use $d = \frac{5T}{K}$, which put emphasis on the prior, and allow adaptation that do not deviate too much from the covariance priors.

5.6 Evaluation set up

The evaluation of the VFOA models was conducted using the IHPD database (Section 5.2). Below, we describe our performance measures and give details about the experimental protocol.

5.6.1 Performance Measures

We propose two kinds of error measures for performance evaluation.

The frame based recognitionrate (FRR):
The frame based recognition rate corresponds to the percentage of frames, or equivalently, the proportion of time, during which the VFOA has been correctly recognized. This rate, however, can be dominated by long duration VFOA events (a VFOA event is defined as a temporal segment with the same VFOA label). Since we are also interested in the temporal patterns followed by the VFOA events, which contain information related to interaction, we also need a measure reflecting how well these events, short or long, are recognized.

Event based precision/recall, and F-measure:
Let us consider two sequences of VFOA events: the GT sequence G obtained from human annotation, and the recognized sequence R obtained through the VFOA estimation process. The GT sequence is defined as $G = (G_i = (l_i, I_i = [b_i, e_i]))_{i=1,...N_G}$ where N_G is the number of events in the ground truth G, $l_i \in \mathcal{F}$ is the ith VFOA event label, b_i and e_i the beginning and end time instants of the event G_i. The recognized sequence R is defined similarly. To compute the performance measures, the two sequences are first aligned using a string alignment procedure that takes into account the temporal extent of the events. More precisely, the matching distance between two events G_i and R_j is defined as:

$$d(G_i, R_j) = \begin{cases} 1 - F_I & \text{if } l_i = l_j \text{ and } I_\cap = I_i \cap I_j \neq \emptyset \\ 2 & \text{otherwise (i.e. events do not match)} \end{cases} \tag{5.16}$$

$$\text{with} \qquad F_I = \frac{2\rho_I \pi_I}{\rho_I + \pi_I}, \quad \rho_I = \frac{|I_\cap|}{|I_i|}, \quad \pi_I = \frac{|I_\cap|}{|I_j|} \tag{5.17}$$

where $|.|$ denotes the cardinality operator, and F_I measures the degree of overlap between two events. Then, given the alignment we can compute for each person, the recall ρ_E, the precision π_E, and the F-measure F_E measuring the events recognition performances and defined as:

$$\rho_E = \frac{N_{matched}}{N_G} \quad , \quad \pi_E = \frac{N_{matched}}{N_R} \quad \text{and} \quad F_E = \frac{2\rho_E \pi_E}{\rho_E + \pi_E} \tag{5.18}$$

where $N_{matched}$ represents the number of events in the recognized sequence that match the same event in the GT after alignment. According to the definition in Eq. 5.16, events are said to match whenever their common intersection is not empty (and labels match). Thus, one may think that the counted matches could be generated by spurious accidental matches due to very small intersection. In practice, however,

acronyms	description
gt	the head pose measurements are the ground truth data obtained with the magnetic sensor
tr	the head pose measurements are those obtained with the head tracking algorithm
gmm	the VFOA model is a GMM
hmm	the VFOA model is an HMM
ML	maximum likelihood approach: the meeting used for testing is used to train the model parameters
p	VFOA priors (π for GMM, A for HMM) learnt from data
ge	parameters of the Gaussian were set using the geometric gaze approach
ad	VFOA model parameters were adapted

Table 5.2: Model acronyms: combinations of acronyms describe which experimental conditions are used. For example, gt-gmm-ge-ad specifies an adapted VFOA GMM model applied to ground truth pose data where the Gaussian parameters before adaptation were given by the geometric gaze model.

Model parameters	
μ_i, Σ_i	Gaussian parameters - learned (μ_i^l, Σ_i^l) or given by geometric modeling (μ_i^g, Σ_i^g), cf Subsection 5.4.3 and 5.4.4.
π, A	GMM and HMM model priors - learnt or set by hand to 'uniform' values π^u, A^u, cf Subsection 5.4.3.
$\kappa_\alpha, \kappa_\beta$	gaze factors - set by hand.
Adaptation hyper-parameters	
ν	scale factor of Dirichlet distribution - set through cross-validation.
$w_i, b_{i,j}$	Dirichlet prior values of π_i and $a_{i,j}$ - set to π_i^u and $a_{i,j}^u$.
τ	scale factor of Normal prior distribution on mean - set through cross-validation.
m_i	VFOA mean prior value of Normal prior distribution - set to either μ_i^l or μ_i^g.
d	scale factor of Wishart prior distribution on covariance matrix - set by hand (cf Sec. 5.5.4).
V_i	VFOA covariance matrices prior values in Wishart distribution - set to either $(d-2)\Sigma_i^l$ or $(d-2)\Sigma_i^g$.

Table 5.3: VFOA Modeling parameters: description and setting.

we observe that it is not the case: the vast majority of matched events have a significant degree of overlap, as illustrated in Fig. 5.9, with 90% of the matches exhibiting an overlap higher than 50%, even with the noisier tracking data.
In Eq. 5.18, the recall measures the percentage of ground truth events that are correctly recognized while the precision measure the percentage of estimated events that are correct. Both precision and recall need to be high to characterize a good VFOA recognition performance. The F-measure, defined as the harmonic mean of recall and precision, reflects this requirement. Finally, the performance measures reported over the whole database (for each seating position) are the average of the precision, recall and F-measure F_E of the 8 individuals.

5.6.2 Experimental protocol

To study the different modeling aspects, several experimental conditions have been defined. They are summarized in Table 5.2 along with the acronyms that identify them in the result tables. Besides, a summary of all parameters involved in the modeling is displayed in Table 5.3.
First, there are two alternatives regarding the head pose measurements: the ground truth *gt* case, where the data are those obtained using the FOB magnetic field sensor, and the *tr* case which relies on the estimates obtained with the tracking system described in Section 5.3. Secondly, there are the two VFOA models, *gmm* and *hmm*, as described in Subsections 5.4.1 and 5.4.2.
Regarding learning, the default protocol is the leave-one-out approach: each meeting recording is in turn

data	ground truth (gt)					tracking estimates (tr)				
modeling	ML	gmm	gmm-p	hmm	hmm-p	ML	gmm	gmm-p	hmm	hmm-p
FRR	79.7	72.3	74.8	72.3	72.5	57.4	47.3	51.3	47.4	48.2
recall	79.6	72.6	69.6	65.5	65.3	66.4	49.1	44.8	38.4	37.6
precision	51.2	55.1	56.2	66.7	66.5	28.9	30	39.5	59.3	60.1
F-measure F_E	62	62.4	61.9	65.8	65.6	38.2	34.8	39.3	45.2	45.3

Table 5.4: VFOA recognition results for person left under different experimental conditions (see Table 5.2).

data	ground truth (gt)					tracking estimates (tr)				
modeling	ML	gmm	gmm-p	hmm	hmm-p	ML	gmm	gmm-p	hmm	hmm-p
FRR	68.9	56.8	61.6	57.3	61.6	43.6	38.1	49.1	38	38.3
recall	72.9	66.6	65.1	58.4	58.2	65.6	55.9	48.7	37.3	37.4
precision	47.4	49.9	51.4	63.5	64.1	24.1	26.8	35.2	55.1	55.9
F-measure F_E	56.9	54.4	55.8	59.5	59.7	34.8	35.6	40.4	43.8	44.2

Table 5.5: VFOA recognition results for person right under different experimental conditions (see Table 5.2).

left aside for testing, while the data of the 7 other recordings are used for parameter learning, including hyper-parameter selection in the adaptation case (denoted *ad*). The maximum likelihood case *ML* is an exception, in which the training data for a given meeting recording is composed of the same single recording. Also, by default, the prior model parameters π or A are set to their 'uniform' values π^u or A^u, as discussed in Subsection 5.4.3. If these parameters are actually learned from the training data, this will be specified with a *p* in the result tables. Note that in the adaptation case, the hyper-parameters of the prior distribution on these parameters are always set according to the 'uniform' values. As for the *ge* acronym, it denotes the case where the VFOA Gaussian means and covariances were set according to the geometric model described in Subsection 5.4.4 instead of being learned from training data. Finally, the adaptation hyper-parameter pair (ν, τ) was selected (in the cartesian set $C^\nu \times C^\tau$) by cross-validation over the training data, using F_E as performance measure to maximize.

5.7 Experimental results

This section provides results under the different experimental conditions. We first analyze the results obtained using ground truth (GT) data, discussing the effectiveness of the modeling w.r.t. different issues (relevance of head pose to model VFOA gaze targets, predictability of VFOA head pose parameters, influence of priors). Secondly, we compare the results obtained with the tracking estimates with those obtained with the GT data. Then, we comment the results of the adaptation scheme, and finally, we examine the results obtained using the geometric modeling. In all cases, results are given separately for the left and right persons (see Fig. 5.1). Some result illustrations are provided in Fig.5.7.

5.7.1 Results exploiting the GT head pose data

In this section. the head pose measurements are given by the flock-of-birds magnetic sensors.

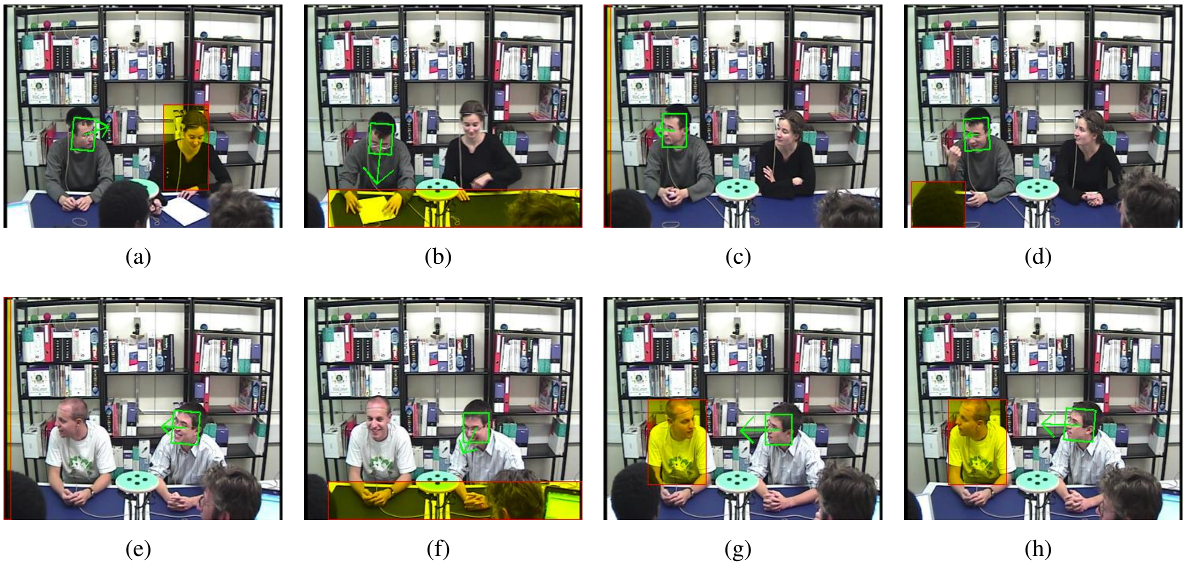
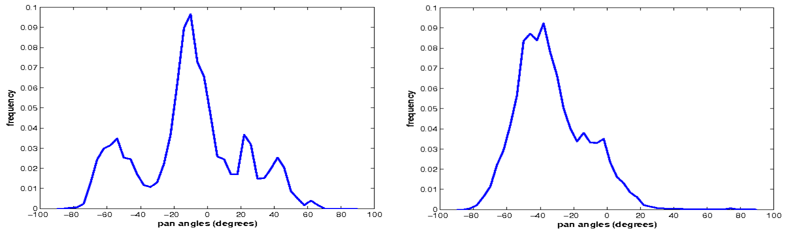


Figure 5.7: Example of results and focus ambiguity. In green, tracking result and head pointing direction. In yellow, recognized focus (hmm-ad condition). Images g and h: despite the high visual similarity of the head pose, the true focus differ (in g: PL; in h: SS). Resolving such cases can only be done by using context (speaking status, other's people gaze, slide activity etc).

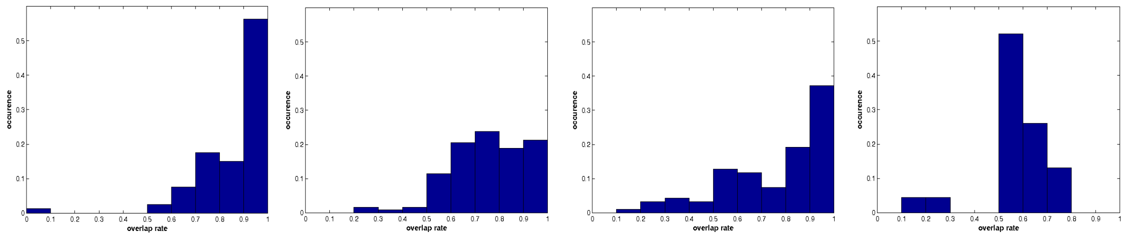
VFOA and head pose correlation:
Table 5.4 and 5.5 display the VFOA recognition results for person left and right respectively. The first column of these two tables gives the results of the ML estimation (see Tab. 5.2) with a GMM modeling. These results show, in an optimistic case, the performances our model can achieve, and illustrate somehow the correlation between a person's head poses and his VFOA. As can be seen, this correlation is quite high for person left (almost 80% FRR), showing the good concordance between head pose and VFOA. This correlation, however, drops to near 69% for person right. This can be explained by the fact that for person right, there is a strong ambiguity between looking at PL or SS, as illustrated by the empirical distributions of the pan angle in Fig. 5.8. Indeed, the range of pan values within which the three other meeting participants and the slide screen VFOA targets lies is almost half the pan range of the person left. The average angular distance between these targets is around 20°for person right, a distance which can easily be covered using only eye movements rather than head pose rotation. The values of the confusion matrices, displayed in Figure 5.10, corroborate this analysis. The analysis of Tables 5.4 and 5.5 shows that this discrepancy holds for all experimental conditions and algorithms, with a performance decrease from person left to person right of approximately 10-13% and 6% for FRR and event F-measure respectively.

VFOA prediction:
In the ML condition, very good results are achieved but they are biased because the same data are used for training and testing. On the contrary, the GMM and HMM results in Table 5.4 and 5.5, for which



(a) Left person pan distribution (b) Right person pan distribution

Figure 5.8: Empirical distribution of the GT head pan angle computed over the database for person left (5.8(a)) and person right (5.8(b)). For person left, the people and slide screen VFOA targets can still be identified through the pan modes. For person right, the degree of overlap is quite significant.

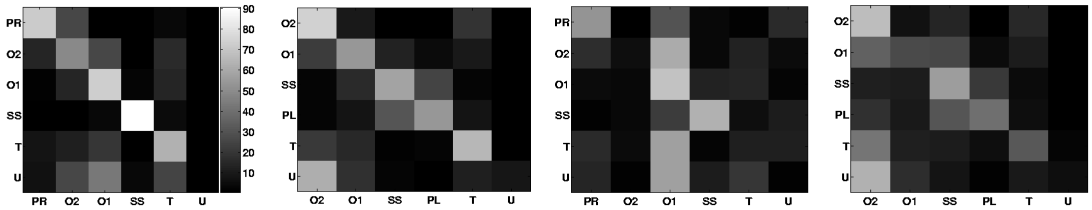


(a) GT, Left (b) GT, Right (c) TR, Left (d) TR, Right

Figure 5.9: Distribution of overlap measures F_I between true and estimated matched events. The estimated events were obtained using the HMM approach. GT and TR respectively denote the use of GT head pose data and tracking estimates data. Left and Right denote person left and right respectively.

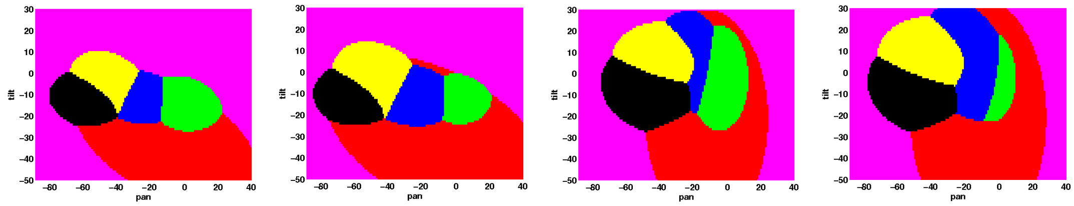
the VFOA parameters are learned from other persons' data, show the generalization property of the modeling. We can notice that the GMM and HMM modeling with or without a prior term produce results close to the ML case. For both person left and right, the GMM approach is achieving better frame recognition and event recall performance while the HMM is giving better event precision and F_E results. This can be explained since the HMM approach is somehow denoising the event sequence. As a result some events are missed (lower recall) but the precision increases due to the elimination of short spurious detections.

VFOA confusions:
Figure 5.10(a) and 5.10(b) display as images the confusion matrices for person left and right obtained with the VFOA FRR performance measure and an HMM modeling. They clearly exhibit confusion between near VFOA targets. For instance, for person left, $O2$ is sometimes confused with PR or $O1$. For person right, the main source of confusion is between PL and SS, as already mentioned. In addition, the table TB can be confused with $O1$ and $O2$, as can be expected since these targets share more or less the same pan values with TB. Thus, most of the confusion can be explained by the geometry of the room



(a) GT, Left (b) GT, Right (c) TR, Left (d) TR, Right

Figure 5.10: Frame-based recognition confusion matrices obtained with the HMM modeling (gt-hmm and tr-hmm conditions). VFOA targets 1 to 4 have been ranked according to their pan proximity: PR: person right - PL: person left - O1 and O2: organizer 1 and 2 - SS: slide screen - TB: table - U: unfocused. Columns represent the recognized VFOA.



(a) GT, without prior (b) GT, with prior (c) TR, without prior (d) TR, with prior

Figure 5.11: Pan-tilt space VFOA decision maps for person right built from all meetings, in the GMM case (cf Eq. 5.3), using GT or tracking head pose data, and with/without VFOA priors. Black=PL, yellow=SS, blue=$O1$, green=$O2$, red=TB, magenta =U.

and the fact that people can modify their gaze without modifying their head pose, and therefore do not always need to turn their head to focus on a specific VFOA target.

Influence of priors:
Table 5.4 and 5.5 also present the recognition rates when learning the prior on the events (-*p* extension). While the improvement w.r.t. no prior is moderate using the GT head pose data or the HMM modeling, it is quite beneficial in the GMM case applied to tracking pose estimates. The effect of the prior is illustrated in Fig. 5.11. While the VFOA target $O2$ has its decision area reduced, $O1$ sees its decision surface extended because people look more often at $O1$ in our database. In practice, the prior allows to clearly favor most likely VFOA target while making the recognition of less likely target more difficult. Although results show that the use of priors can improve performance, their usage could clearly be a problem when using the VFOA recognition system on other unrelated meetings (or if $O1$ and $O2$ would have exchanged seats across meetings). Thus, in the remaining, we will not use such prior in the experiments.

Comparison with other algorithms:
We can compare our performance to the recent work [114]. In this paper, an interesting study on VFOA interaction modeling has been conducted, where one of the task, among others, consisted in estimating the VFOA of four people engaged in a conversation, using people's speaking status (annotated manually) and head pose measured with magnetic field sensors. For each person, the VFOA targets were the three other participants. The authors of [114] report an average FRR of 67.9 %. Despite the lower number of VFOA targets and the use of multiple cues (speech and head pose), their results are similar to ours. We obtained 57% for person right and 72.3% for person left using the HMM recognizer (resp. 62% and 72.7% with adaptation, as shown later).

5.7.2 Results with head pose estimates

Table 5.4 and 5.5 provide the results obtained using the head pose tracking estimates, under the same experimental conditions than when using the GT head pose data. As can be seen, substantial performance degradation can be noticed. In the ML case, the decrease in FRR and F-measure ranges from 22% to 26% for both person left and right. These degradations are mainly due to small pose estimation errors and also, sometimes, large errors due to short periods when the tracker locks on a subpart of the face. Fig. 5.11 illustrates the effect of pose estimation errors on the VFOA distributions. The shape changes in the VFOA decision maps when moving from GT pose data to pose estimates conveys the increase of pose variance measured for each VFOA target. The increase is moderate for the pan angle, but quite important for the tilt angle.
A more detailed analysis of Table 5.4 and 5.5 shows that the performance decrease (from GT to tracking data) in the GMM condition follows the ML case, while the deterioration in the HMM case is smaller, in particular for F_E. This demonstrates that, in contrast with what was observed with the clean GT pose data, in presence of noisy data, the HMM smoothing effect is quite beneficial. Also, the HMM performance decrease is smaller for person right (19% and 15% for respectively FRR and F_E) than for person left (25% and 20%). This can be due to the better tracking performance -in particular regarding the pan angle- achieved on people seated at the person right position (cf Table 5.1). Figure 5.12 presents the plot of the VFOA FRR versus the pan angle tracking error for each meeting participant, when using GT head pose data (i.e. with no tracking error) or pose estimates. It shows that for person left, there is a strong correlation between tracking error and VFOA performance, which can be due to the fact that higher tracking errors directly generate larger overlap between VFOA class-conditional pose distributions (cf Fig. 5.8, left). For person right, this correlation is weaker, as the same good tracking performance result in very different VFOA recognition results. In this case, the higher level of inherent ambiguities between several VFOA targets (e.g. SS and PL) may play a larger role.
Finally, the 2 right images of Figure 5.10(c) and 5.10(d) display the confusion matrices when using the HMM model and the head pose estimates. The same confusion than using the GT data are exhibited, but more pronounced because of the tracking errors (see above) and tilt estimation uncertainties.

5.7.3 Results with model adaptation

Tables 5.6 and 5.7 display the recognition performance obtained with the adaptation framework described in Section 5.5[3]. For person left, one can observe no improvement when using GT data and a large im-

[3] In the tables, we recall the values without adaptation for ease of comparison.

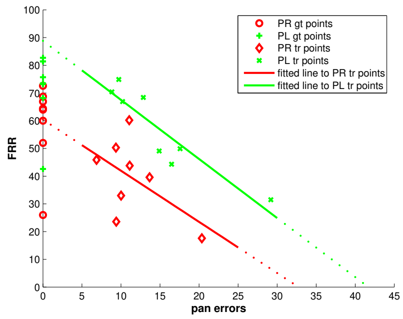


Figure 5.12: VFOA frame based recognition rate vs head pose tracking errors (for the pan angle), plotted per meeting. The VFOA recognizer is the HMM modeling after adaptation.

error measure	gt-gmm	gt-gmm-ad	gt-hmm	gt-hmm-ad	tr-gmm	tr-gmm-ad	tr-hmm	tr-hmm-ad
FRR	72.3	72.3	72.3	72.7	47.3	57.1	47.4	53.1
recall	72.6	72.1	65.5	68.8	49.1	48.7	38.4	40.5
precision	55.1	53.7	66.7	64.4	30	41	59.3	62.5
F-measure F_E	62.4	61.2	65.8	66.2	34.8	42.8	45.2	47.9

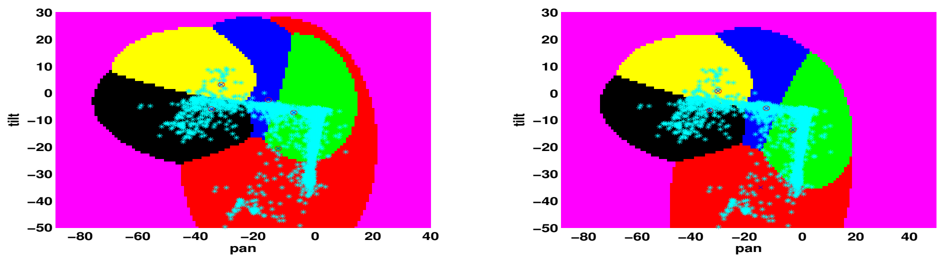
Table 5.6: VFOA recognition results for person left, before and after adaptation.

provement when using the tracking estimates (e.g. around 10% and 8% for resp. FRR and F_E with the GMM model). In this situation, the adaptation is able to cope with the tracking errors and the variability in looking at a given target. For person right, we notice improvement with both the GT and tracking head pose data. For instance, with the HMM model and tracking data, the improvement is 3.8% and 5% for FRR and F_E. Again, in this situation adaptation can cope for people's personal way of looking to the targets, such as correcting the bias in head tilt estimation, as illustrated in Figure 5.13.
When exploring the optimal adaptation parameters estimated through cross-validation, one obtain the histograms of Figure 5.14. As can be seen, regardless of the kind of input pose data (GT or estimates), they correspond to configurations giving approximately equal balance to the data and prior regarding the adaptation of the HMM transition matrices (ν_1 and ν_2), and configurations for which the data are driving the adaptation process of the mean pose values (τ_1 and τ_2).

Comparison with other algorithms:
Our results, 42% FRR for person right and 53% for person left, are quite far from the 73% reported in the interesting paper [102]. Several factors may explain the difference. First, in [102], a 4 people meeting situation was considered and no other VFOA target apart from the other meeting participants was considered. In addition, these participants were sitting at equally spaced positions around a round table, optimizing the discrimination between VFOA targets. Besides, people were recorded from a camera placed in front of them. Hence, due to the table geometry, the very large majority of head pan lay between $[-45, 45]$ degrees, where the tracking errors are smaller (see Table 5.1) Ultimately, our results are more in accordance with the 52% FRR reported by the same authors [125] when using the same

error measure	gt-gmm	gt-gmm-ad	gt-hmm	gt-hmm-ad	tr-gmm	tr-gmm-ad	tr-hmm	tr-hmm-ad
FRR	56.8	59.3	57.3	62	38.1	39.3	38	41.8
recall	66.6	70.2	58.4	63	55.9	55.3	37.3	43.6
precision	49.7	50.9	63.5	64.5	26.8	29	55.1	56.1
F-measure F_E	54.4	56.4	59.5	62.7	35.6	37.3	43.8	48.8

Table 5.7: VFOA recognition results for person right, before and after adaptation.



(a) Before adaptation

(b) After adaptation

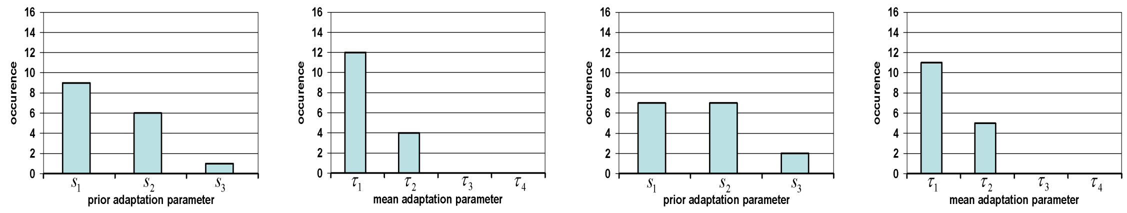
Figure 5.13: VFOA decision map example before adaptation and after adaptation. After adaptation, the VFOA of $O1$ and $O2$ correspond to lower tilt values. black=PL, yellow=SS, blue=$O1$, green=$O2$, red=TB, magenta =U. The blue stars represent the tracking head pose estimates used for adaptation.

framework [102] but applied to a 5-person meeting, resulting in 4 VFOA targets.

5.7.4 Results with the geometric VFOA modeling

Here we report the results obtained when setting the model parameters by exploiting the meeting room geometry, as described in Subsection 5.4.4. This possibility for setting parameters is interesting because it removes the need for data annotation each time a new VFOA target is considered, e.g. when people are moving around in the room.

Figure 5.15 shows the geometric VFOA Gaussian parameters (mean and covariance) generated by the model when using $(\kappa_\alpha, \kappa_\beta) = (0.5, 0.5)$. The parameters $(\kappa_\alpha, \kappa_\beta)$ were defined in Section 5.4.4. As can be seen, the VFOA pose values predicted by the model are consistent with the average pose values computed for individuals using the GT pose data, especially for person left. For person right, we can observe that the geometric model is wrongly predicting the gaze value of $O2$ (not moved) and $O1$ (attraction in the wrong direction). Indeed, for person right, our assumption that the reference head orientation N in Fig. 5.4 consists of looking at the other side of the table is not appropriate: as all the VFOA targets are located on their right side, people tend to already orient their shoulder towards their right as well (see Figure 5.5(c)), and thus N should be set accordingly. Assuming more logically that the reference looking direction corresponds to looking at $O1$, we obtain a better match. This is demonstrated by Table 5.8,



(a) GT data (b) GT data (c) TR data (d) TR data

Figure 5.14: Histogram of the optimal scale adaptation factor of the HMM prior, and HMM VFOA mean (b and d), selected though cross-validation on the training set, and when working with GT head pose data or with tracking head pose estimates.

Method	learned VFOA		geometric VFOA	
Error	E_{pan}	E_{tilt}	E_{pan}	E_{tilt}
person left	6.37	5.08	5.54	6.35
person right (reference direction: looking straight (at $O2$)	5.85	6.07	12.5	7.65
person right (reference direction: looking at $O1$)	5.85	6.07	5.62	7.65

Table 5.8: Prediction errors for learned VFOA and geometric VFOA models (with GT pose data).

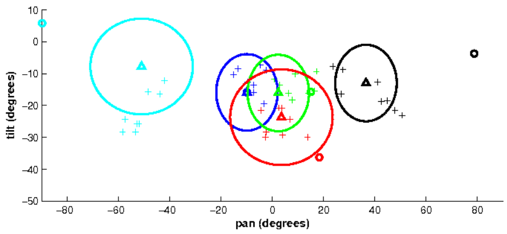
which provides the prediction errors in pan E_{pan} defined as:

$$E_{pan} = \frac{1}{8 \times (K-1)} \sum_{m=1}^{8} \sum_{f_i \in \mathcal{F}/\{U\}} |\bar{\alpha}_m(f_i) - \alpha_m^p(f_i)| \tag{5.19}$$

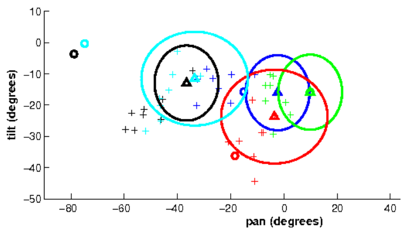
where $\bar{\alpha}_m(f_i)$ is the average pan value of the person in meeting m and for the VFOA f_i, and $\alpha_m^p(f_i)$ is the predicted value according to the chosen model (i.e. the pan component of $\mu_{f_i}^g$ or $\mu_{f_i}^l$ in the geometric or learning approaches respectively[4]). The tilt prediction error E_{tilt} is obtained by replacing pan angles by tilt angles in Equation 5.19.

The recognition performance are presented in Tables 5.9 and 5.10. For person right, the model based on $O1$ as reference looking direction N is used. These tables show that, when using GT head pose data, the results are slightly worse than with the learning approach, which is somehow in accordance with the similarity in the prediction errors. However, when using the pose estimates, the results are better. For instance, for person left, with adaptation, the FRR improvement is of more than 6%. Given that the modeling does not request any training data (except for camera calibration, and the currently manual setting of the reference direction N), this is an interesting and encouraging result. Also, we can notice that the adaptation always improve the recognition, sometimes quite significantly (see the GT data condition for person right, or the tracking data for person left).

[4]The reported error in the geometric case follows the same scheme than in the learning case: for each meeting the employed κ_α has been learned on the other meetings.



(a) Left person



(b) Right person

Figure 5.15: Geometric VFOA Gaussian distributions for person left and person right: the figure displays the gaze target direction ($\bigcirc$), the corresponding head pose contribution according to the geometric model with values $(\kappa_\alpha, \kappa_{tilt}) = (0.5, 0.5)$ ($\triangle$ symbols), and the average head pose (from GT pose data) of individual people ($+$). Ellipses display the standard deviations used in the geometric modeling. black=PL or PR, cyan=SS, blue=$O1$, green=$O2$, red=TB.

Measure	gt	gt-ge	gt-ad	gt-ge-ad	tr	tr-ge	tr-ad	tr-ge-ad
FRR	72.3	69.3	72.3	70.8	47.4	55.2	53.1	59.5
recall	72.1	61.4	68.8	64.4	38.4	42	40.5	41.9
precision	55.1	70.2	64.4	67.3	59.3	63.7	62.5	69.9
F-measure F_E	61.2	65.2	66.6	65.3	45.2	48.2	47.9	50.1

Table 5.9: VFOA recognition results for person left using the HMM model with geometric VFOA parameter setting ($(\kappa_\alpha, \kappa_{tilt}) = (0.5, 0.5)$), with/without adaptation.

5.8 Conclusion

In this chapter we presented a methodology to recognize the VFOA of meeting participants from their head pose, the latter being defined by its pan and tilt angles. Such head pose measurements were obtained either through magnetic field sensors or using a probabilistic based head pose tracking algorithm. The experiments showed that, depending on people's position in the meeting room and on the angular distribution of the VFOA targets, the eye gaze may or may not be highly correlated with the head pose.

In absence of such correlation, and if eye gaze tracking is unaccessible due to low resolution images, the way to improve VFOA recognition can only come from the prior knowledge embedded in the cognitive and interactive aspects of human-to-human communication. Ambiguous situations, such as the one illustrated in Fig. 5.7(g) and 5.7(h), where the same head pose can correspond to two different VFOA targets, could be resolved by the joint modeling of the speaking and VFOA characteristics of all meeting participants. Such characteristics have been shown to exhibit specific patterns/statistics in the behavioral and cognitive literature, as already exploited by [114].

Besides, as shown by the experiments, there indeed exists some correlation between head pose tracking errors and VFOA recognition results. Improving the tracking algorithms, e.g. using multiple cameras, higher resolution images or adaptive appearance modeling techniques, would thus improve the VFOA results. Finally, in the case of meetings in which people are moving to the slide screen or white board for presentations, the development of a more general approach that models the VFOA of these moving

measure	gt	gt-ge	gt-ad	gt-ge-ad	tr	tr-ge	tr-ad	tr-ge-ad
FRR	57.3	51.8	62	58.5	38	41.1	41.8	42.7
recall	58.4	43.7	63	52.2	37.3	41.9	43.6	43.8
precision	63.5	69	64.5	71.5	55.1	61.1	56.1	61.1
F-measure F_E	59.5	53	62.7	59.2	43.8	49.1	48.8	50.1

Table 5.10: VFOA recognition results for person right using the HMM model with geometric VFOA parameter setting $((\kappa_\alpha, \kappa_{tilt}) = (0.5, 0.5))$, with/without adaptation.

people will be necessary. Investigations about modeling the VFOA of moving people is the topic of the following chapter.

Chapter 6

Wandering visual focus of attention (VFOA) recognition

In the previous chapter, we studied the VFOA of static person in meeting contexts. In this chapter, in collaboration with Smith *et al* [126] we address the more general case of recognizing the VFOA of moving persons (but with static VFOA targets w.r.t. the camera). This is an interesting task with many applications. For instance, let us consider a situation where an advertising firm has been asked to produce an outdoor display ad campaign for use in shopping malls and train stations. Internally, the firm has developed several competing designs, one of which must be chosen to present to the client. Analysis of the VFOA of people exposed to the advertisement can be used to judge the best placement and content of these outdoor advertisements, such does for television the Nielsen's ratings which measures media effectiveness by estimating the size of the net cumulative audience of a program through surveys and Nielsen Boxes [127]. If one were to design an automatic Nielsen-like system for outdoor display advertisements, it might automatically determine the number of people who have actually viewed the advertisement as a percentage of the total number of people exposed to it. This is an example of what we have termed the wandering visual focus of attention (WVFOA) problem, in which the tasks are:

1. to automatically detect and track an unknown, varying number of people able to move about freely,
2. and to estimate their visual focus of attention (VFOA).

The WVFOA problem is an extension of the traditional VFOA [102] problem in the sense that in WVFOA, mobility is unconstrained. As a result, the subject's target of attention may be mobile, or as the subject moves about the scene, his appearance may change as he attempts to keep his attention on a specific target. Unconstrained motion also limits the resolution of the subject (see 6.1), as a wide field of view is necessary to capture multiple subjects over an area of interest. In the example of the outdoor advertisement application, the goal is to identify each person exposed to the advertisement and determine if they looked at it. Additionally, we can collect other useful statistics such as the amount of time they spent looking at the advertisement.

In this thesis, our goal is to propose a principled probabilistic framework for estimating WVFOA for moving people. We applied our method to the advertising example to demonstrate its usefulness in real-life applications. Our solution assumes a fixed camera which can be placed arbitrarily, so long as the subjects appear clearly within the field of view. Our method requires a training phase in which the

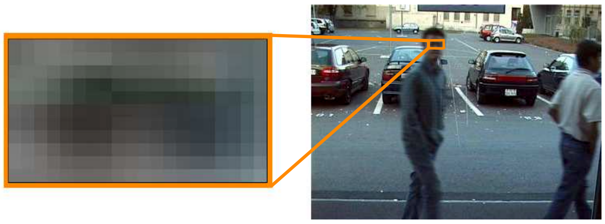


Figure 6.1: In the WVFOA problem, allowing the unconstrained motion of multiple people complicates the task of estimating a subject's visual focus of attention (VFOA). Here, the resolution is too low to estimate his focus of attention from eye gaze.

appearance of people in the scene (and the orientation of their heads) is modeled. Our method consists of two parts: a dynamic Bayesian network (DBN), which simultaneously tracks people in the scene and estimates their head pose, and a WVFOA model, which infers a subject's VFOA from their location and head pose.

Besides defining the WVFOA problem itself, first, we propose a principled probabilistic framework for solving the WVFOA problem by designing a mixed-state DBN that jointly represents the people in the scene and their various parameters. The state-space is formulated in a true multi-person fashion, consisting of size and location parameters for the head and body, as well as head pose parameters for each person in the scene. This type of framework facilitates defining interactions between people.

Because the dimension of the state representing a single person is sizable, the multi-object state-space can grow to be quite large when several people appear together in the scene. Efficiently inferring a solution in such a framework can be a difficult problem. We present an efficient method to do inference in this high-dimensional model. This is done using a trans-dimensional Markov Chain Monte Carlo (MCMC) sampling technique, an efficient global search algorithm robust to the problem of getting trapped in local minima.

Then, we demonstrate the real-world applicability of our model by applying it to the outdoor advertisement problem described earlier. We show that we are able to gather useful statistics such as the number of viewers and the total number of people exposed to the advertisement.

Finally, we thoroughly evaluate our model in the context of the outdoor advertisement application using realistic data and a detailed set of objective performance measures.

The remainder of the Chapter is organized as follows. In the next Section we discuss related works. In Section 6.2, we describe our joint multi-person and head-pose tracking model. In Section 6.3, we present a method for modeling a person's VFOA. In Section 6.4 we describe our procedure for learning and parameter selection. In Section 6.5 we test our model on captured video sequences of people passing by an outdoor advertisement and evaluate its performance. Finally, Section 6.6 contains some concluding remarks.

6.1 Related work

To our knowledge, this work is the first attempt to estimate the WVFOA for moving people as defined in this thesis. However, there is an abundance of literature concerning the three component tasks of the WVFOA problem: multi-person tracking, head pose tracking, and estimation of the visual focus of attention (VFOA). Related work to head tracking and VFOA recognition have been discussed in Chapter 2 and 5. In the following we discuss work related to multi-object tracking.

Multi-person tracking is the process of locating a variable number of moving persons or objects over time. Multi-person tracking is a well studied topic, and a multitude of approaches have been proposed. Here, we will restrict our discussion to probabilistic tracking methods which use a particle filter (PF) formulation [128, 129, 130, 131, 132]. Some computationally inexpensive methods use a single-object state-space model [131], but suffer from the inability to resolve the identities of different objects or model interactions between objects. As a result, much work has been focused on adopting a rigorous Bayesian joint state-space formulation to the problem, where object interactions can be explicitly defined [128, 129, 130, 133, 134, 132, 126]. However, sampling from a joint state-space can quickly become inefficient as the dimension of the space increases when more people or objects are added [128]. Recent work has concentrated on using MCMC sampling to track multiple people more efficiently. In [133] ants were tracked using MCMC sampling and a simple observation model. In [134], multiple humans were tracked from overhead as they crossed a college campus. Smith et *al*, in [126], extended this model to handle varying number of people using a reversible-jump MCMC sampling technique. In this thesis we extend the model of [126] by handling a much more complex object models and a larger state-space. This has necessitated the non-trivial design of new jump types and proposal distributions, as well as inter- and intra-personal interactions (see Section 6.2.4).

6.2 Joint multi-person and head-pose tracking

In this section, we first recall the basis of Bayesian tracking formulation with the MCMC approach and then describe the different and specific models that we used in the context of the WVFOA study. In a Bayesian approach to multi-person tracking, the goal is to estimate the conditional probability for joint multi-person configurations of people $\mathbf{X}_t$, taking into account a sequence of observations $z_{1:t} = (z_1, ..., z_t)$. In our model, a joint multi-person configuration, or joint state, is the union of the set of individual states describing each person in the scene. The observations consist of information extracted from an image sequence. The posterior distribution $p(\mathbf{X}_t|z_{1:t})$ is expressed recursively by

$$p(\mathbf{X}_t|z_{1:t}) \quad = \quad C^{-1}p(z_t|\mathbf{X}_t) \times \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|z_{1:t-1})d\mathbf{X}_{t-1}, \tag{6.1}$$

where the motion model, $p(\mathbf{X}_t|\mathbf{X}_{t-1})$, governs the temporal evolution of the joint state $\mathbf{X}_t$ given the previous state $\mathbf{X}_{t-1}$, and the observation likelihood, $p(z_t|\mathbf{X}_t)$, expresses how well the observed features z_t fit the predicted state $\mathbf{X}_t$. Here C is a normalization constant.

In practice computing the filtering distribution using (6.1) is often intractable. As already said in Sections 2 and 4, it can be approximated by applying the Monte Carlo method, in which the target distribution (Eq. 6.1) is represented by a set of N samples $\{\mathbf{X}_t^{(n)}, n = 1, ..., N\}$, where $\mathbf{X}_t^{(n)}$ denotes the n-th sample. For efficiency, in this work we use the Markov Chain Monte Carlo (MCMC) method because in high dimensional space, MCMC is more efficient than other sampling methods such as importance sampling. In MCMC, the set of samples have equal weights and form a so-called Markov chain. According

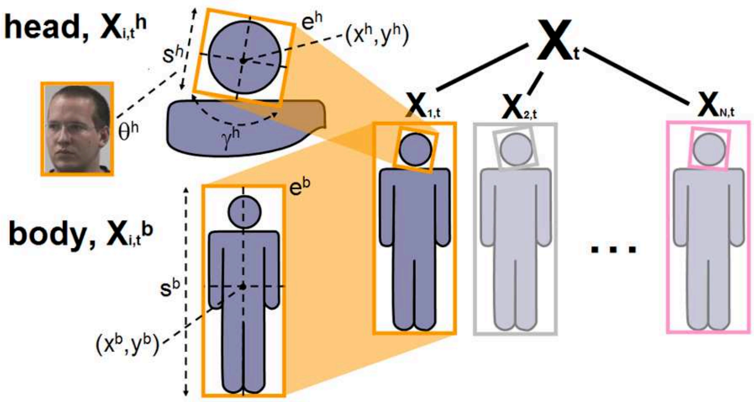


Figure 6.2: State model for varying numbers of people. The joint multi-person state, X_t consists of an arbitrary number of people $X_{i,t}$, each of which contain a body $\mathbf{X}^b_{i,t}$ and head $\mathbf{X}^h_{i,t}$ component. The body is modeled as a bounding box with parameters for the location (x^b, y^b), height scale s^b, and eccentricity e^b. The head location L^h has similar parameters for location (x^h, y^h), height s^h, and eccentricity e^h, as well as in-plane rotation r^h. The head also has an associated exemplar k^h, which models the out-of-plane head rotation.

to Section 4.4, given the samples at the previous step, an approximation of Eq. 6.1 is written

$$p(\mathbf{X}_t|z_{1:t}) \approx C^{-1} p(z_t|\mathbf{X}_t) \sum_n p(\mathbf{X}_t|\mathbf{X}^{(n)}_{t-1}). \tag{6.2}$$

Our goal is now to build a chain a Markov chain of sample which represents the filtering distribution. Following this approach, the remainder of Section 6.2 is devoted to describing the details of our joint multi-person and head-pose tracking model. In the next sub-Section, we will discuss exactly how we model an individual person and the set of multiple people in the scene. In Section 6.2.2 we explain how to model motion and interactions between people. We describe our observation likelihood model, which estimates how well a proposed configuration fits the observed data, in Section 6.2.3. In Section 6.2.4, we discuss how to form the Markov Chain representing the distribution of Eq. 6.2 using the Metropolis-Hastings (MH) algorithm, and in Section 6.2.5 we show how to infer a point estimate solution from the posterior distribution.

6.2.1 State model for a varying number of people

The state at time t describes the joint multi-object configuration of people in the scene. Because the amount of people in the scene may vary, we explicitly use a state model designed to accommodate changes in dimension [126], instead of a model with a fixed dimension, as in [133]. The joint state vector $\mathbf{X}_t$ is defined by $\mathbf{X}_t = \{X_{i,t}|i \in \mathcal{I}_t\}$, where $X_{i,t}$ is the state vector for person i at time t, and $\mathcal{I}_t$ is the set of all person indexes. The total number of people present in the scene is $m_t = |\mathcal{I}_t|$, where $|\cdot|$ indicates set cardinality. A special case exists when there are no people present in the scene, denoted by $\mathbf{X}_t = \emptyset$.

In our model, each person is represented by two components: a body $\mathbf{X}^b_{i,t}$, and a head $\mathbf{X}^h_{i,t}$ as seen in Figure 6.2. Note that we drop the i and t subscripts for the remainder of this section for simplicity. The body component is represented by a bounding box, whose state vector contains four parameters, $\mathbf{X}^b = (x^b, y^b, s^b, e^b)$. The point (x^b, y^b) is the continuous 2D location of the center of the bounding box in the image, s^b is the height scale factor of the bounding box relative to a reference height, and e^b is the eccentricity defined by the ratio of the width of the bounding box over its height.

The head component of the person model is represented by a bounding box which may rotate in the image plane, along with an associated discrete exemplar used to represent the head-pose (see Section 6.2.3 for more details). The state vector for the head is defined by $\mathbf{X}^h = (L^h, k^h)$ where $L^h = (x^h, y^h, s^h, e^h, r^h)$ denotes the continuous 2D configuration of the head, including the continuous 2D location (x^h, y^h), the height scale factor s^h, the eccentricity e^h, and the in-plane rotation r^h. A discrete variable, k^h represents the head-pose exemplar which models the out-of-plane head rotation. See Chapter 4 for more details about the head state.

6.2.2 Dynamics and interaction

The dynamic model governs the evolution of the state between time steps. It is responsible for predicting the motion of people (and their heads) as well as governing transitions between the head-pose exemplars. It is also responsible for modeling inter-personal interactions between the various people, as well as intra-personal interactions between the body and the head. The overall dynamic model for multiple people is written

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) \stackrel{def}{=} p_V(\mathbf{X}_t|\mathbf{X}_{t-1})p_0(\mathbf{X}_t), \tag{6.3}$$

where p_V is the predictive distribution responsible for updating the state variables based on the previous time step, and p_0 is a prior distribution modeling interactions.

To model the multi-person predictive distribution, we follow the approach of [126] where p_V is defined as

$$p_V(\mathbf{X}_t|\mathbf{X}_{t-1}) = \prod_{i\in\mathcal{I}_t} p(\mathbf{X}_{i,t}|\mathbf{X}_{t-1}), \tag{6.4}$$

when people are present in the previous time step ($\mathbf{X}_{t-1} \neq \emptyset$), and constant otherwise. However, in this work, the motion for a single person i, $p(\mathbf{X}_{i,t}|\mathbf{X}_{t-1})$, is dependent only on its own previous state $p(\mathbf{X}_{i,t}|\mathbf{X}_{i,t-1})$ if that person existed in the previous frame. If not, $p(\mathbf{X}_{i,t}|\mathbf{X}_{t-1})$ is formed from a 2D initialization distribution, $p_{init}(\mathbf{X}_{i,t})$, formed by smoothing the foreground segmented image of the previous frame as seen in Figure 6.3.

The motion model for a single person is given by

$$p(\mathbf{X}_{i,t}|\mathbf{X}_{i,t-1}) = p(\mathbf{X}^b_{i,t}|\mathbf{X}^b_{i,t-1})p(L^h_{i,t}|L^h_{i,t-1})p(\theta^h_{i,t}|\theta^h_{i,t-1}), \tag{6.5}$$

where the dynamics of the body state $\mathbf{X}^b_i$ and the head spatial state component L^h_i are modeled as 2^{nd} order auto-regressive (AR) processes. The dynamic of the head-pose exemplars dynamic θ^h_i is a transition table learned from training data (see Chapter 4).

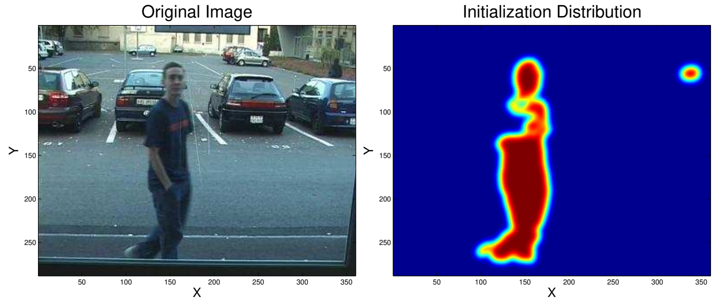


Figure 6.3: The initialization distribution in (center) determines where to place a new person in the scene by applying a Gaussian smoothing filter to the foreground segmentation of the original image (left).

The interaction model $p_0(\mathbf{X}_t)$ handles two types of interactions, inter-personal p_{0_1} and intra-personal p_{0_2}: $p_0(\mathbf{X}_t) = p_{0_1}(\mathbf{X}_t)p_{0_2}(\mathbf{X}_t)$. For modeling inter-personal interactions, we follow the method proposed in [133]. In this method, the inter-personal interaction model $p_{0_1}(\mathbf{X}_t)$ serves the purpose of restraining trackers from fitting the same person by penalizing overlap between trackers. This is achieved by exploiting a pairwise Markov Random Field (MRF) whose graph nodes are defined at each time step by the people, and the links by the set $\mathcal{C}$ of pairs of proximate people. By defining an appropriate potential function

$$\phi(\mathbf{X}_{i,t}, \mathbf{X}_{j,t}) \propto \exp(-g(\mathbf{X}_{i,t}, \mathbf{X}_{j,t})), \tag{6.6}$$

the interaction model

$$p_{0_1}(\mathbf{X}_t) = \textstyle\prod_{ij\in\mathcal{C}} \phi(\mathbf{X}_{i,t}, \mathbf{X}_{j,t}) \tag{6.7}$$

enforces a constraint in the multi-person dynamic model, based on locations of a person's neighbors. This constraint is defined by a non-negative penalty function, g, which is based on the spatial overlap of pairs of people (g is null for no overlap, and increases as the area of overlap increases).

We also introduce intra-personal interactions to the overall motion model. The intra-personal interaction model is meant to constrain the head model w.r.t. the body model, so that they are configured in a physically plausible way (i.e. the head is not detached from the body, or located near the waist). The intra-personal interaction model $p_{0_2}(\mathbf{X}_t)$ is defined as

$$p_{0_2}(\mathbf{X}_t) = \prod_{k\in\mathcal{I}_t} p(L^h_{k,t}|\mathbf{X}^b_{k,t}), \tag{6.8}$$

and penalizes head configurations which fall outside of an accepted domain defined by the configuration of the body. The penalization increases when the center of the head falls further than a distance outside of the top third of the body bounding box.

With these terms defined, the Monte Carlo approximation of the overall tracking filtering distribution in Eq. 6.2 can now be expressed

$$\begin{aligned} p(\mathbf{X}_t|z_{1:t}) &\approx C^{-1}p(z_t|\mathbf{X}_t)p_0(\mathbf{X}_t)\sum_n p_V(\mathbf{X}_t|\mathbf{X}_{t-1}^{(n)}) \\ &\approx C^{-1}p(z_t|\mathbf{X}_t)\prod_{ij\in\mathcal{C}}\phi(\mathbf{X}_{i,t},\mathbf{X}_{j,t})\prod_{k\in\mathcal{I}_t}p(L_{k,t}^h|\mathbf{X}_{k,t}^b)\sum_n p_V(\mathbf{X}_t|\mathbf{X}_{t-1}^{(n)}). \end{aligned} \tag{6.9}$$

6.2.3 Observation model

The observation model estimates the likelihood of a proposed configuration, or how well the proposed configuration is supported by evidence from the observed features. Our observation model consists of a body model and a head model, formed from a set of five total features. The body model consists of binary and color features, which are global in that they are defined pixel-wise over the entire image. The binary features (z_t^{bin}) make use of a foreground segmented image, while color features (z_t^{col}) exploit histograms in hue-saturation (HS) space. The head model is local in that its features (z^h) are gathered independently for each person. They are responsible for the localization of the head and estimation of the head-pose, and include texture z_t^{tex}, skin color z_t^{skin}, and silhouette z_t^{sil} features. For the remainder of this Section, the time index (t) has been omitted to simplify notation. Assuming conditional independence of body and head observations, the overall likelihood is given by

$$p(z|\mathbf{X}) \triangleq p(z^{col}|z^{bin},\mathbf{X})p(z^{bin}|\mathbf{X})p(z^h|\mathbf{X}), \tag{6.10}$$

where the first two terms constitute the body model and the third term represents the head model. The body model, the head model, and each of the five component features are detailed in the following subsections.

Body model

The body observation model is responsible for detecting and tracking people, adding or removing people from the scene, and maintaining consistent identities. It is comprised of a binary feature and a color feature.

Body binary feature:
The binary feature is responsible for tracking bodies, and adding and removing people from the scene. The binary feature relies on an adaptive foreground segmentation technique described in [135]. At each time step, the image is segmented into sets of foreground pixels $\mathcal{F}$ and background pixels B from the images ($I = \mathcal{F} \cup B$), which form the foreground and background observations ($z^{bin,\mathcal{F}}$ and $z^{bin,B}$)

For a given multi-person configuration and foreground segmentation, the binary feature computes the distance between the observed overlap (between the area of the multi-person configuration $S^{\mathbf{X}}$ obtained by projecting $\mathbf{X}$ onto the image plane and the segmented image) and a learned value. Qualitatively, we are following the intuition of a statement such as: "We have observed that two well-placed trackers (tracking two people) should contain approximately 65% foreground and 35% background." The overlap is measured for F and B in terms of precision ν and recall ρ : $\nu^{\mathcal{F}} = \frac{S^{\mathbf{X}}\cap\mathcal{F}}{S^{\mathbf{X}}}$, $\rho^{\mathcal{F}} = \frac{S^{\mathbf{X}}\cap\mathcal{F}}{\mathcal{F}}$, $\nu^B = \frac{S^{\mathbf{X}}\cap B}{S^{\mathbf{X}}}$, and $\rho^B = \frac{S^{\mathbf{X}}\cap B}{B}$. An incorrect location or person count will result in ν and ρ values that do not match the learned values well, resulting in a lower likelihood and encouraging the model to choose better multi-person configurations.

The binary likelihood is computed for the foreground and background case

$$p(z^{bin}|\mathbf{X}) \triangleq p(z^{bin,\mathcal{F}}|\mathbf{X})p(z^{bin,B}|\mathbf{X}) \tag{6.11}$$

where the definition of the binary foreground term, $p(z^{bin,\mathcal{F}}|\mathbf{X})$, for all non-zero person counts ($m \neq 0$) is a single Gaussian distribution in precision-recall space ($\nu^{\mathcal{F}}$,$\rho^{\mathcal{F}}$). The binary background term, on the other hand, is defined as a set of Gaussian mixture models (GMM) learned for each possible person count ($m \in \mathcal{M}$). For example, if the multi-person state hypothesizes that two people are present in the scene, the binary background likelihood term is the GMM density of the observed ν^B and ρ^B values from the GMM learned for $m = 2$. For details on the learning procedure, see Section 6.4.

Body Color Feature:
The color feature is responsible for maintaining the identities of people over time, as well as assisting the binary feature in localization of the body. The color feature uses HS color observations from the segmented foreground and background regions ($z^{col,\mathcal{F}}$ and $z^{col,B}$) . Assuming conditional independence between foreground and background, the color likelihood is written

$$p(z^{col}|z^{bin}, \mathbf{X}) = p(z^{col,\mathcal{F}}|z^{bin,\mathcal{F}}, \mathbf{X})p(z^{col,B}|z^{bin,B}, \mathbf{X}). \tag{6.12}$$

The first term (foreground color likelihood) determines how well the color of each measured person matches online learned models, and the second term (background color likelihood) determines how well the background matches an off-line learned background model.

The foreground color likelihood compares an extracted 4D multi-person color histogram to an adaptive learned model, by

$$p(z^{col,\mathcal{F}}|z^{bin,\mathcal{F}}, \mathbf{X}) \propto e^{\lambda_{\mathcal{F}} d^2_{\mathcal{F}}}, \tag{6.13}$$

where $d_{\mathcal{F}}$ is the Bhattacharya distance (see Equation 2.9) between the learned model and observed histogram and $\lambda_{\mathcal{F}}$ is a parameter. The adaptive model is chosen from a small set of adaptive foreground color models (multiple models allow slightly different color models to compete). The model histograms are defined over a person index i, spatial segment (roughly corresponding to the head, torso, and legs), hue (H), and saturation (S) color spaces (quantized into 8 bins). Every frame, a vote is cast for the model that best matches the extracted data, and a 'winning' adaptive model is chosen from the set based on the number of 'votes' it has collected (the 'winner' is used in the above likelihood expression).

The background color likelihood helps reject configurations with untracked people by penalizing unexpected colors (i.e. those found on a person). The background model is a static 2D HS color histogram, learned from empty training images. The background color likelihood is defined as

$$p(z_t^{col,B}|z_t^{bin,B}, \mathbf{X}_t) \propto e^{\lambda_B d^2_B}, \tag{6.14}$$

where λ_B and d^2_B are defined as in the foreground case.

Head model

The head model is responsible for localizing the head and estimating the head-pose. The head likelihood is defined as

$$p(z^h|\mathbf{X}) = \left[\prod_{i\in\mathcal{I}} p_{tex}(z_i^{tex}|\mathbf{X}_i)p_{skin}(z_i^{skin}|\mathbf{X}_i)p_{sil}(z_i^{sil}|\mathbf{X}_i).\right]^{\frac{1}{m}}. \quad (6.15)$$

The individual head likelihood terms are geometrically averaged by $\frac{1}{m}$ to balance the overall likelihood as the number of people, m, varies. This non-standard likelihood normalization is needed to make the likelihood for different number of person comparable. The head model is obtained from three features: texture z_t^{tex}, skin color z_t^{skin}, and silhouette z_t^{sil}. The feature extraction procedure and likelihood functions are the same as the one described in Chapter 4.

6.2.4 Trans-dimensional MCMC

In this sub-section, we describe how Reversible-Jump Markov Chain Monte Carlo (RJMCMC) can be used to efficiently generate a Markov Chain which represents the distribution of Equation 6.9. In Section 6.2.5, we will show how to infer a solution to the tracking problem from the Markov Chain, and in Section 6.2.6 we give an overview of the algorithm.

The multi-person state-space can quickly become very large when we allow for an arbitrary number of people. The state vector for a single person is ten-dimensional. Traditional sequential importance resampling (SIR) particle filters are inefficient in such high-dimensional spaces [136]. Markov chain Monte Carlo (MCMC) particle filters are more efficient, but do not allow for the dimensionality of the state-space to vary (fixing the number of people). To solve this problem, we have adopted the RJMCMC sampling scheme, originally proposed in [134] which retains the efficiency of MCMC sampling but allows for "reversible jumps" which can vary the dimensionality of the state-space.

Constructing a Markov chain with Metropolis-Hastings:

As previously mentioned, the stationary distribution of the Markov Chain must be defined over the configuration space $\mathbf{X}_t$, it must be able to vary in dimension, and it must approximate the filtering distribution defined in Eq. 6.9 (please note that for the remainder of this section, we omit the time subscript t for simplicity). For the task of constructing the Markov Chain, we use the Metropolis-Hastings (MH) algorithm [136].

Starting from an arbitrary initial configuration[1] $\mathbf{X}$, the MH algorithm samples a new configuration $\mathbf{X}^*$ from a proposal distribution $Q(\mathbf{X}^*|\mathbf{X})$, and adds the proposed sample to the Markov Chain with probability

$$a = \min\left(1, \frac{p(\mathbf{X}^*)Q(\mathbf{X}|\mathbf{X}^*)}{p(\mathbf{X})Q(\mathbf{X}^*|\mathbf{X})}\right), \quad (6.16)$$

otherwise, the current configuration $\mathbf{X}$ is added to the Markov Chain. A Markov Chain is constructed by repeatedly adding samples to the chain in this fashion.

[1] In this work, we initialize the Markov chain at time t by sampling uniformly an initial state from the Markov chain at $t-1$.

Here, in practice, in contrary to what was done with the MCMC head tracker, a new configuration $\mathbf{X}^*$ is chosen by first selecting a *move type*, υ^* from a set of reversible moves Υ (defined in the next subsection) with prior probability p_{υ^*}. Moves must be defined in such a way that every move type which changes the dimensionality of the state has a corresponding reverse move type [137]. The acceptance ratio a can be re-expressed through *dimension-matching* [137] as

$$a = \min\left(1, \frac{p(\mathbf{X}^*)p_\upsilon Q_\upsilon(\mathbf{X})}{p(\mathbf{X})p_{\upsilon^*}Q_{\upsilon^*}(\mathbf{X}^*)}\right), \tag{6.17}$$

where Q_υ is a move-specific distribution, defined for all move types (except swap) as

$$Q_\upsilon(\mathbf{X}_t^*) = \sum_i Q_\upsilon(i)Q_\upsilon(\mathbf{X}_t^*|i), \tag{6.18}$$

over all people i, in such a way that the move is applied to a target index i^*, while the rest of the multi-person configuration is fixed.

Splitting the six reversible move types:

We define six move types in our model: birth, death, swap, body update, head update, and pose update. In previous works using RJMCMC [126, 138], a single update move was defined in which all the parameters of a randomly selected person were updated simultaneously. This was sufficient for less complex object models, but a problem arises when multiple features are used to evaluate different aspects of the state. For example, let us imagine an update move designed to apply motion to the body, apply motion to the head, and adjust the head-pose simultaneously. Even if applying such a move results in a better overall likelihood, there is no guarantee that the individual body, head, and pose configurations have improved. Some may have remained the same, or even worsened. Because the ranges of the terms of the overall likelihood vary, some will dominate. The result might be a model which tracks bodies well, but poorly estimates the head-pose (which we observed in practice). We refer to this as the likelihood balancing problem.

To overcome this problem, we propose to decouple the update move into three separate moves: body update, head update, and pose update. In this way, we divide the task of finding a good configuration for an entire person into three smaller problems: finding a good configuration for the body, for the head location, and for the head-pose.

We now define the six move types. For each move type, we explain how to choose a target index, and define the move-specific proposal distributions and acceptance ratios.

(1) Birth. A new person i^* is added to the new configuration $\mathbf{X}^*$ which was not present in the old configuration $\mathbf{X}$: ($\mathcal{I}_t^* = \mathcal{I}_t \cup \{i^*\}$). This implies a dimension change from m_t to $m_t + 1$.

To add a new person, a birth location, x_b^*, is sampled from the birth proposal distribution described in Figure 6.3 using a stratified sampling strategy. Next, a new person index, or target index, must be chosen, which may be a previously 'dead' person, or an 'unborn' person. The target index i^* is sampled from the birth target proposal model, $Q_{birth}(i)$, which is defined in such a way that the probability of choosing a 'dead' person depends on their temporal distance and spatial distance to the sampled birth location (recently killed, nearby people are most likely to be reborn). The probability of choosing an 'unborn' person is unity minus the sum of 'dead' probabilities.

Having chosen a target index, the birth move is applied to i^* while the rest of the multi-person configuration is fixed; in Eq. 6.18 this is done by defining $Q_\upsilon(\mathbf{X}_t^*|i)$ as

$$Q_{birth}(\mathbf{X}_t^*|i) = \frac{1}{N}\sum_n p(\mathbf{X}_{i,t}^*|\mathbf{X}_{t-1}^{(n)}) \prod_{l\in\mathcal{I}_t} p(\mathbf{X}_{l,t}|\mathbf{X}_{t-1}^{(n)})\delta(\mathbf{X}_{l,t}^* - \mathbf{X}_{l,t}). \tag{6.19}$$

Initial body parameters of a new object (born or reborn) are sampled from learned Gaussian distributions. Initial head and pose parameters are chosen to maximize the head likelihood.

Now that the identity and parameters of the new person i^* have been determined, it can be shown that the acceptance ratio for the new multi-person configuration $\mathbf{X}_t^*$ is given by

$$a_{birth} = \min\left(1, \frac{p(z_t|\mathbf{X}_t^*)\prod_{j\in\mathcal{C}_{i^*}}\phi(\mathbf{X}_{i^*,t}^*, \mathbf{X}_{j,t}^*)}{p(z_t|\mathbf{X}_t)}\frac{p_{death}}{p_{birth}}\frac{Q_{death}(i^*)}{Q_{birth}(i^*)}\right), \tag{6.20}$$

where $\mathcal{C}_{i^*}$ and ϕ are pairs of proximate objects and the interaction potential defined in Section 6.2.2, and $q_{death}(i^*)$ is the reverse-move death target proposal model. Note that the interaction model helps discourage births that overlap existing people and complexity is reduced as many terms in $p(z_t|\mathbf{X}_t^*)$ and $p(z_t|\mathbf{X}_t)$ cancel.

(2) Death. An existing person i^* is removed from the new configuration $\mathbf{X}^*$ which was present in the old configuration $\mathbf{X}$ ($\mathcal{I}_t^* = \mathcal{I}_t \backslash \{i^*\}$) where $\backslash$ is the difference between sets. This implies a dimension change from m_t to $m_t - 1$. The target index i^* is chosen with probability $q_{death}(i)$ by uniformly sampling from the set of 'live' people (i.e. $q_{death}(i) = \frac{1}{m_t}$). Person i^* is removed keeping the rest of the multi-person configuration fixed, with mixture components defined as

$$Q_{death}(\mathbf{X}_t^*|i) = \frac{1}{N}\sum_n \prod_{l\in\mathcal{I}_t, l\neq i} p(\mathbf{X}_{l,t}|\mathbf{X}_{t-1}^{(n)})\delta(\mathbf{X}_{l,t}^* - \mathbf{X}_{l,t}), \tag{6.21}$$

and the acceptance probability can shown to simplify to

$$a_{death} = \min\left(1, \frac{p(z_t|\mathbf{X}_t^*)}{p(z_t|\mathbf{X}_t)\prod_{j\in\mathcal{C}_{i^*}}\phi(\mathbf{X}_{i^*,t}, \mathbf{X}_{j,t})}\frac{p_{birth}}{p_{death}}\frac{Q_{birth}(i^*)}{Q_{death}(i^*)}\right). \tag{6.22}$$

(3) Swap. The configurations of a pair of objects, i^* and j^* are swapped. The proposal is a mixture model over pairs of objects $Q_s(\mathbf{X}_t^*) = \sum_{i,j} q_{swap}(i,j)Q_{swap}(\mathbf{X}_t^*|i,j)$. Candidates are chosen with probability $q_{swap}(i,j)$, which is defined such that the probability a pair of people are chosen is a function of their proximity (nearby pairs are more likely to be selected). When the move is applied, the mixture component $Q_{swap}(\mathbf{X}_t^*|i,j)$ swaps the configurations of objects i^* and j^*. It can be shown that the acceptance ratio for the swap move is reduced to

$$a_{swap} = \min\left(1, \frac{p(z_t|\mathbf{X}_t^*)}{p(z_t|\mathbf{X}_t)}\right). \tag{6.23}$$

(4) Body update. The body parameters $\mathbf{X}_{i,t}^b$: including its location (x^b, y^b), height s^b, and eccentricity

e^b are updated. The body update move proposal is defined as $Q_{body}(\mathbf{X}^*) = \sum_i \frac{1}{m_t} Q_{body}(\mathbf{X}^*|i)$ with

$$\begin{aligned} Q_{body}(\mathbf{X}^*|i) &= \frac{1}{N}\sum_n p(\mathbf{X}^{b,*}_{i^*,t}|\mathbf{X}^{(n)}_{t-1}) p(\overline{\mathbf{X}^{b,*}_{i^*,t}}|\mathbf{X}^{(n)}_{t-1}) \delta(\overline{\mathbf{X}^{b,*}_{i^*,t}} - \overline{\mathbf{X}^{b}_{i^*,t}}) \prod_{l \in \mathcal{I}_t \setminus i^*} p(\mathbf{X}_{l,t}|\mathbf{X}^{(n)}_{t-1}) \delta(\mathbf{X}^*_{l,t} - \mathbf{X}_{l,t}) \\ &= \frac{1}{N}\sum_n w^{b,*,(n)}_{i^*,t} p(\mathbf{X}^{b,*}_{i^*,t}|\mathbf{X}^{(n)}_{t-1}) \end{aligned} \tag{6.24}$$

where $\overline{\mathbf{X}^b_{i^*,t}}$ denotes all state parameters except $\mathbf{X}^b_{i^*,t}$, and $\mathbf{X}^{b,*}_{i^*,t}$ denotes the proposed body configuration for target i^*. In practice, this implies first selecting a person randomly, i^*, and sampling a new body configuration for this person from $p(\mathbf{X}^{b,*}_{i^*,t}|\mathbf{X}^{b,n^*}_{t-1})$, using a particle n^*, sampled according to the weights $w^{b,*,(n)}_{i^*,t}$, from the previous time and keeping all the other parameters unchanged. With this proposal, the acceptance probability a_{body} can then be shown to reduce to:

$$a_{body} = \min\left(1, \frac{p(z^b_t|\mathbf{X}^{b,*}_{i^*,t}) p(L^{h,*}_{i^*,t}|\mathbf{X}^{b,*}_{i^*,t}) \prod_{j \in \mathcal{C}_{i^*}} \phi(\mathbf{X}^*_{i^*,t}, \mathbf{X}^*_{j,t})}{p(z^b_t|\mathbf{X}^{b}_{i^*,t}) p(L^{h}_{i^*,t}|\mathbf{X}^{b}_{i^*,t}) \prod_{j \in \mathcal{C}_{i^*}} \phi(\mathbf{X}_{i^*,t}, \mathbf{X}_{j,t})}\right). \tag{6.25}$$

(5) Head update. This move implies sampling the new head spatial configuration of person i^* in a similar fashion according to $p(L^*_{i^*,t}|L^{n^*}_{t-1})$. The acceptance ratio a_{head} simplifies to

$$a_{head} = \min\left(1, \frac{p(z^{h,*}_{i^*,t}|\mathbf{X}^{h,*}_{i^*,t}) p(L^{h,*}_{i^*,t}|\mathbf{X}^{b,*}_{i^*,t})}{p(z^{h}_{i^*,t}|\mathbf{X}^{h}_{i^*,t}) p(L^{h}_{i^*,t}|\mathbf{X}^{b}_{i^*,t})}\right). \tag{6.26}$$

(6) Pose update. The last move consists of simply sampling the new head-pose from the proposal function $p(\theta^*_{i^*,t}|\theta^{n^*}_{t-1})$ and accepting with probability a_{pose}:

$$a_{pose} = \min\left(1, \frac{p(z^{h}_{i^*,t}|\mathbf{X}^{h,*}_{i^*,t})}{p(z^{h}_{i^*,t}|\mathbf{X}^{h}_{i^*,t})}\right). \tag{6.27}$$

6.2.5 Inferring a solution

In RJMCMC, the first N_b samples added to the Markov Chain (using the MH algorithm) are part of the burn-in cycle, which allows the Markov Chain to reach the target density. The filtering distribution is approximated by the N_p samples taken after the burn-in point. The Markov Chain, however, does not provide a single answer to the tracking problem.

For this reason, we compute a point estimate, which is a single multi-person configuration calculated from the stationary distribution that serves as the tracking output. To determine the (discrete) configuration of people in the scene, we search for the most common configuration of people, taking into account swapped identities, births, and deaths. Using these samples, we determine the (continuous) body $\mathbf{X}^b_{i,t}$ and head spatial configurations $L^h_{i,t}$ for the various people in the scene (including head roll $\gamma_{i,t}$) by taking the Marginal Mean of each parameter. For the out-of-plane head rotations represented by the discrete exemplar θ_i, we take the Marginal Mean of the corresponding Euler angles for pan and tilt.

6.2.6 Pseudo-code

The detailed steps of our joint multi-person body-head tracking and WVFOA estimation model is summarized in Figure 6.4.

6.3 WVFOA modeling

The WVFOA task is to automatically detect and track a varying number of people able to move about freely, and to estimate their VFOA. The WVFOA problem is significantly more complex than the traditional VFOA problem because it allows for a variable number of moving people instead of a single stationary person. The advertising application we have chosen as an introduction to WVFOA represents a relatively simple instance of the problem because we are only attempting to measure the focus of attention on a single target: the advertisement. More complex WVFOA scenarios could have several targets, moving targets, or both.

For the advertising application, a person's WVFOA is defined as being in one of two states:

- focused - looking at the advertisement, or
- unfocused - not looking at the advertisement.

Note that this is just one of many ways in which the WVFOA can be represented, but it is sufficient to solve the task targeted in this work. A persons state of focus depends both on their location and on their head-pose as seen in Figure 6.5. For head location and head-pose information, we rely on the output of the RJMCMC tracker described in Section 6.2.

To model the WVFOA, we chose to check for only the focused state, though this method could be easily extended to model both focused and unfocused states. To determine if a person is in a focused state, we extract the pointing vector z^h from the pose estimate output by the RJMCMC tracker, which is characterized by the pan and tilt angles, as well as the horizontal head position x^h (see Figure 6.5). Because the target advertisement is stationary, the ranges of z^h corresponding to the focused state are directly dependent on the location of the head in the image. For this reason, we chose to split the image into $K = 5$ horizontal regions $I_k, k = \{1, ..., 5\}$, and modeled the likelihood of a focused state as

$$p(z^h) = \sum_{k=1}^{K} p(x^h \in I_k, z^h) = \sum_{k=1}^{K} p(x^h \in I_k) p(z^h | x^h \in I_k) \tag{6.28}$$

where the first term $p(x^h \in I_k)$ models the likelihood a person's head location belongs to region I_k, and the second term $p(z^h|x^h \in I_k)$ models the likelihood of focused head-pose given the region the head belongs to. The inclusion of the head location in modeling the WVFOA allowed us to solve an issue not previously addressed [114, 117, 102]: resolving the WVFOA of a person whose focused state depends on their location.

The terms of the WVFOA model in Equation 6.28 are defined as follows. The image horizontal axis, x, is divided into K regions I_k whose centers and width are denoted by x_{I_k} and σ_{I_k}, respectively. The probability of a head location x^h belonging to region I_k is modeled by a Gaussian distribution $p(x^h \in I_k) = \mathcal{N}(x^h; x_{I_k}, \sigma_{I_k})$. For each region, the distribution of pointing vectors representing a

Algorithm 1: Multi-Person Body/Head Tracking and WVFOA Estimation with RJMCMC

At each time step t, the posterior distribution of Eq. 6.9 is represented by a Markov Chain consisting of a set of $N = N_b + N_p$ samples $\{\mathbf{X}_t^{(n)}, n = N_b, ..., N\}$. Body, head, and pose parameters are inferred from the Markov Chain after it reaches the burn-in point. Using these values, the WVFOA model determines if the persons attention is focused on the advertisement or unfocused.

1. `Initialize` the MH sampler by choosing a sample from the $t-1$ Markov Chain with the MPM number of people (m_{t-1}^{MPM}). Apply the motion model and accept it as sample $n = 1$.
2. `Metropolis-Hastings Sampling`. Draw $N = N_b + N_p$ samples according to the following schedule (where N_b is the burn-in point):
   - `Begin` with the state of the previous sample $X_t^{(n)} = X_t^{(n-1)}$.
   - `Choose Move Type` by sampling from the set of moves $\Upsilon = \{$ birth, death, swap, body update, head update, pose update$\}$ with prior probability p_{υ^*}.
   - `Select a Target` i^* (or set of targets i^*, j^* for swap) according to the target proposal $q_\upsilon(i)$ for the selected move type.
   - `Sample New Configuration` $\mathbf{X}^*{}_t$ from the move-specific proposal distribution Q_{υ^*}. For the various move types, this implies:
     - Birth - add a new person i^* $m_t^{(n)*} = m_t^{(n)} + 1$ according to Eq. 6.19.
     - Death - remove an existing person i^* $m_t^{(n)*} = m_t^{(n)} - 1$ according to Eq. 6.21.
     - Swap - swap the parameters of two existing people i^*,j^* $\mathbf{X}_{i,t}^{(n)} \rightarrow \mathbf{X}_{j,t}^{(n)*}$, $\mathbf{X}_{j,t}^{(n)} \rightarrow \mathbf{X}_{i,t}^{(n)*}$.
     - Body Update - update the body parameters $X_{i,t}^{b,(n)*}$ of an existing person i^* (Eq. 6.24).
     - Head Update - update the head parameters $L_{i,t}^{h,(n)*}$ of an existing person i^*.
     - Pose Update - update the pose parameter $k_{i,t}^{(n)*}$ of an existing person i^*.
   - `Compute Acceptance Ratio` α according to Equations 6.20, 6.22, 6.23, 6.25 6.26, and 6.27.
   - `Add` n^{th} `Sample` to the Markov Chain: If $\alpha \geq 1$, then add the proposed configuration ($\mathbf{X}^*$). Otherwise, add the proposed $\mathbf{X}^*$ with probability α. If the proposed configuration is rejected, add the previous $\mathbf{X}$ (i.e. $\mathbf{X}_t^{(n-1)}$).
3. `Compute a Point Estimate Solution` from the Markov Chain (as in Section 6.2.5):
   - determine the most common multi-person configuration $\hat{X}_t$ accounting for births, deaths, and swaps. Collect the samples of this configuration into a set W.
   - determine the body $\hat{X}_t^b$ and head $\hat{L}_t^h$ spatial configurations, and the out-of-plane head rotations for the various people in the scene by computing the Marginal Mean of the parameters over the set W (using Euler decompositions for pan $\hat{\alpha}^h$ and tilt $\hat{\beta}^h$).
4. `Determine the WVFOA` for each person in the scene (as in Section 6.3):
   - determine the likelihood each person is in a focused state from their horizontal head location $\hat{x}_t^h$ and pointing vector $\hat{z}_t^h$ according to Equation 6.28.
   - if the likelihood is greater than a threshold $p(z^h) > T_{wvfoa}$, that person is focused, otherwise he/she is unfocused.

Figure 6.4: Algorithm for joint multi-person body and head tracking and WVFOA estimation with RJMCMC.

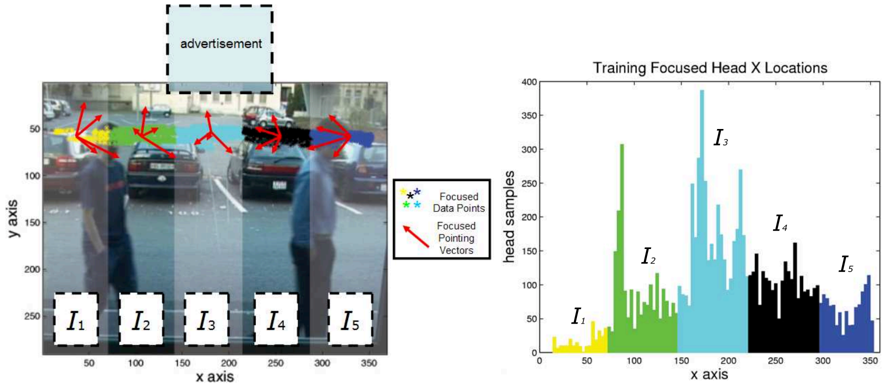


Figure 6.5: WVFOA Modeling. (Left) WVFOA is determined by head-pose and horizontal position in the image. The horizontal axis is split into 5 regions ($I_1, ..., I_5$), and a WVFOA model is defined for each of these regions. Yellow, green, cyan, black, and blue data points represent focused head locations used for training and red arrows represent 2D projections of typical samples of focused pointing vectors z^h. Note that the advertisement is affixed to a window and appears just above the image frame. (Right) Over 9400 training points representing a person in a focused state (also seen in the left pane) were split into 5 regions along the horizontal axis and used to train a Gaussian model for each region.

focused state was modeled using a Gaussian distribution. Typical pointing vectors for each region are seen in Figure 6.5.

The parameters of the WVFOA model (Gaussian mean and covariance matrix) were learned from the training data described in the next section. Though our WVFOA model does not make use of the vertical head location, it is straightforward to generalize the models by allowing other partitions $\{I_k\}$ of the image plane. Finally, a person is determined to be *focused* when the corresponding likelihood $p(z^h)$ in Eq. 6.28 is greater than a threshold, T_{wvfoa}.

As an alternative, one might attempt to bypass the tracking model and find the location and head-pose using a face detector. However, a face detector alone might not be sufficient to solve the WVFOA problem for several reasons: (1) the WVFOA problem allows for a range of head-poses beyond that of typical face detectors (including situations where part or none of the face is visible - partially visible in our case) (2) unless they include an additional tracking stage, existing state-of-the-art face detectors such as that described in [32] have no mechanism to maintain identity between time steps or recover from occlusions. Properties of face detection and tracking are necessary to solve the WVFOA problem, and indeed elements of our head model share commonalities with face detectors.

6.4 Training and parameter selection

In this section we will describe our training procedure and how we determine the key parameters of our model. We begin by describing the setup of the experiment.

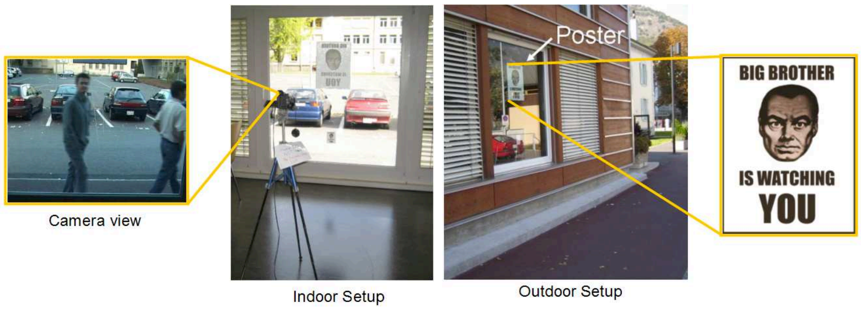


Figure 6.6: *Experimental Setup.* (Left) Inside the building, a camera is placed so that it is facing out a window. The view from the camera can be seen in the insert. (Right) Outside, the advertisement in the window is noticeable to people walking along the footpath. The fake advertisement poster can be seen in the insert.

6.4.1 Experimental setup

To simulate the advertising application, an experiment was set up as seen in Figure 6.6. A fake advertisement was placed in an exposed window with a camera set behind. The camera view can be seen in the left-hand insert, with the bottom edge of the poster appearing at the top of the image above the heads of the subjects.

In our experiments, actors were used due to privacy concerns for actual passers-by. The actors were instructed to pass in front of the window with the freedom to look at the advertisement (or not) as they would naturally. A recording of 10-minute duration (360×288 resolution, 25 frames per sec.) was made in which up to three people appear in the scene simultaneously. The recorded data includes several difficult tracking events such as people passing and occluding each other. Though simulated, every effort was made to ensure that the data was as fair a representation of a real-life scenario as possible.

6.4.2 Training

The recorded video data was organized into a training and test set of equal size and disjoint from each other. The training set, consisting of nine sequences for a total of 1929 frames, was manually annotated for body location, head location, and focused/unfocused state.

The parameters for the foreground segmentation were tuned by hand by observing results on the training set. The binary body feature model was trained with the annotated body locations and foreground segmented binary images of the training set. Using this information, GMMs were trained for precision and recall for the foreground and the background. Head annotations were used to learn the parameters of the Gaussian skin-color distribution in the head-pose skin feature. The silhouette mask was also trained using the head annotations (1929 frames), by averaging the binary patches corresponding to head annotations. Parameters for the WVFOA model, including T_{wvfoa}, were optimized on the training data (bootstrapped to 9400 training points, see Figure 6.5) to achieve the highest WVFOA event recognition performance (see Section 6.5 for details on event recognition performance). The training set was also used to learn prior sizes (scale and eccentricity) for the person models. The head texture and skin color models were learned from the Prima-Pointing Database, which consists of 30 sets of images of 15 people,

Table 6.1: Symbols, values, and descriptions for key parameters of our model.

Parameter	Value	Set by	Description
α_{scale}	0.01	learned	*motion model* body and head scale variance (AR2 process)
$\alpha_{position}$	2.4	learned	*motion model* body and head position variance (AR2 process)
K_{bf}	1	learned	*observation model* body binary model number of Gaussians (foreground)
K_{bb}	3	learned	*observation model* body binary model number of Gaussians (background)
λ_F	20	learned	*observation model* body color foreground hyper-parameter
λ_{sil}	200	learned	*observation model* head silhouette hyper-parameter
λ_{tex}	0.5	learned	*observation model* head texture hyper-parameter
T_{tex}	$\exp(\frac{-9}{2})$	learned	*observation model* head texture threshold
λ_{sk}	0.5	learned	*observation model* head skin color hyper-parameter
p_{birth}	0.05	hand	*RJMCMC* prior probability of choosing a *birth* move
p_{death}	0.05	hand	*RJMCMC* prior probability of choosing a *death* move
p_{swap}	0.05	hand	*RJMCMC* prior probability of choosing a *swap* move
p_{body}	0.283	hand	*RJMCMC* prior probability of choosing a *body update* move
p_{head}	0.283	hand	*RJMCMC* prior probability of choosing a *head update* move
p_{pose}	0.283	hand	*RJMCMC* prior probability of choosing a *pose update* move
N_p	300,600,800	learned	*RJMCMC* number of samples in chain for 1,2,3 simultaneous people, resp.
N_b	0.25*N_p	hand	*RJMCMC* number of *burn-in* samples
K_{wvfoa}	5	hand	*WVFOA model* number of Gaussians
T_{wvfoa}	*0.00095*	learned	*WVFOA model* likelihood threshold

each containing 79 frontal images of the same person in a different pose ranging from -90 degrees to 90 degrees.

6.4.3 Parameter selection

In addition to the trained models, the rest of the parameters of our algorithm were chosen by hand. Some were selected using the training set without exhaustive tuning. Others (e.g. single-person dynamic model parameters) were assigned standard values. Unless explicitly stated, all parameters remain fixed for the evaluation described in the next section. In Table 6.1, a description of the key parameters mentioned in the text and their values are provided.

6.5 Evaluation

As mentioned in the introduction, we applied our model to a hypothetical Nielsen-like outdoor advertisement application. The task was to determine the number of people who actually looked at an advertisement as a percentage of the total number of people exposed to it.

To evaluate the performance of our application, a ground truth for the test set was hand annotated in a similar manner to the training set. The test set consists of nine sequences, a through i. Sequences

Table 6.2: Test set data summary.

sequence	length (s)	# people total	# people simultaneous	# looks at ad	description
a	15	3	1	2	person from right (no look), person from left (looks), person from right (looks)
b	13	3	1	3	person from left (looks), person from right (looks), person from right (looks)
c	10	3	1	3	person from right (looks), person from left (looks), person from right (looks)
d	5	2	2	2	2 people cross from the right, both look at ad
e	6	2	2	3	2 people cross from the left, both look at ad (1^{st} looks twice)
f	4	2	2	2	2 people cross from the left, both look at ad
g	4	2	2	1	2 people cross from the right, 2^{nd} looks at ad
h	4	2	2	2	1 person from right (looks at ad), another from left (no look)
i	11	3	3	4	3 people appear from right, all look at ad (1^{st} looks twice)

a, b, and c contain three people (appearing sequentially) passing in front of the window. Sequences d through h contain two people appearing simultaneously. Sequence i contains three people appearing simultaneously. The details of the test set are summarized in Table 6.2. Our evaluation compared our results with the ground truth over 180 experiments on the 9 test sequences (as our method is a stochastic process, we ran 20 runs per sequence). The length of the Markov Chain was chosen such that there was a sufficient number of samples for good quality tracking according to the number of people in the scene (see Table 6.1). Experimental results are illustrated in Figures 6.7 and 6.11.

In the remainder of this section, we will discuss the performance of the multi-person body and head tracking (Section 6.5.1), the advertisement application (Section 6.5.2), and the effect of varying the length of the Markov chain (Section 6.5.3).

6.5.1 Multi-person body and head tracking performance

To evaluate the multi-person body and head tracking performance we adopt a set of measures proposed in [139]. These measures evaluate three tracking features: the ability to estimate the number and placement of people in the scene (*detection*), the ability to persistently track a particular person over time (*tracking*), and how tightly the estimated bounding boxes fit the ground truth (*spatial fitting*).

To evaluate detection, we rely on the rates of *False Positive* and *False Negative* errors (normalized per person, per frame) denoted by $\overline{FP}$ and $\overline{FN}$. The *Counting Distance* $\overline{CD}$ measures how close the estimated number of people is to the actual number (normalized per person per frame). A $\overline{CD}$ value of zero indicates a perfect match. To evaluate tracking, we report the *Tracker Purity* $\overline{TP}$ and *Object Purity* $\overline{OP}$, which estimate the degree of consistency with which the estimates and ground truths were properly identified ($\overline{TP}$ and $\overline{OP}$ near 1 indicate well maintained identity, near 0 indicate poor performance). For spatial fitting, the *F-measure* measures the overlap between the estimate and the ground truth for the body and head from recall ρ and precision ν, ($F = \frac{2\nu\rho}{\nu+\rho}$). A perfect fit is indicated by $F = 1$, no overlap by $F = 0$.

Per-sequence results appear in Fig. 6.8 with illustrations for sequence f in Fig. 6.7 and for sequences b, e, h, and i in Fig. 6.11. For detection (Fig. 6.8a), the FP and FN rates are reasonably low, averaging a total of 2.0 FN errors and 4.2 FP errors per sequence. These errors usually correspond to

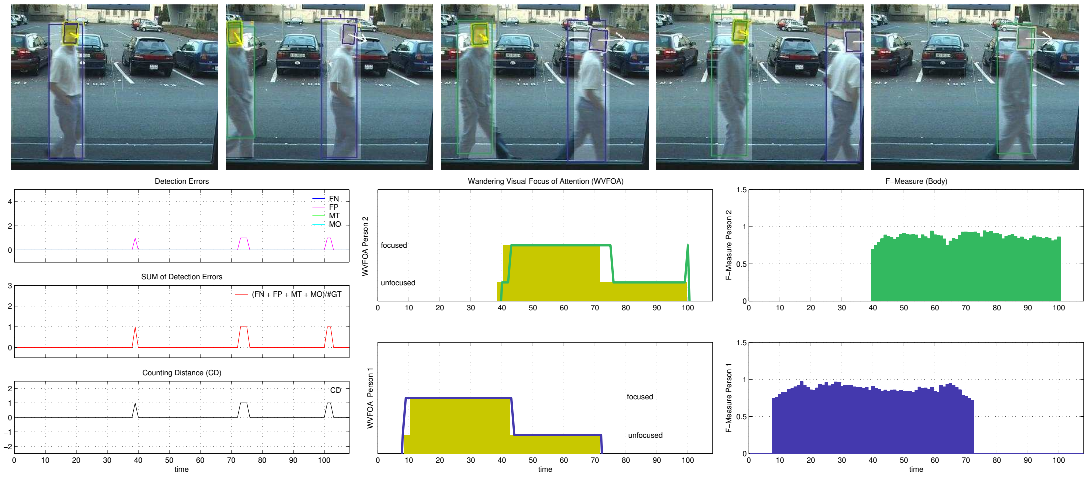


Figure 6.7: *Experimental Results*. Upper Row: Frames from Sequence f in which two people cross the scene from left to right, looking at the advertisement once each. Tracking results appear as green and blue boxes around the body and head (with an associated pointing vector). A yellow pointing vector/head border indicates a *focused* state, a white pointing vector/head border indicates an *unfocused* state. The ground truth appears as shaded boxes for the head and the body (the head area is shaded yellow when labeled as *focused* and grey when labeled as *unfocused*). Bottom Row, Left: The top plot contains a history of person detection errors over the course of the sequence, the middle plot contains a summation over all the errors, the bottom plot shows CD (see text for descriptions of these measures). Center: WVFOA results for both people over the duration of the sequence. The ground truth appears as yellow bars (raised indicates a *focused* state, lowered when *unfocused*, and non-existent when the object does not appear in the scene). The tracking results appear as blue and green lines. Right: F measures how tightly the bounding boxes fit the ground truth for each person.

problems detecting exactly when a person enters or leaves the scene. The overall $\overline{CD}$, which indicates the average error in the estimation of the number of people in the scene, was 0.018 (where zero is ideal).

For tracking (Fig. 6.8b), $\overline{TP}$ and $\overline{OP}$ are both of high quality. Combining $\overline{TP}$ and $\overline{OP}$ using the F-measure as for spatial fitting ($\frac{2\overline{TP}\,\overline{OP}}{\overline{TP}+\overline{OP}}$), we find that overall our model produced a high value (0.93). The main source of error in tracking was due to extraneous trackers appearing when people enter or leave the scene. A second source of error occurred when a person exited the scene followed by another person entering from the same place in a short period of time: the second person was often misinterpreted as the first. Sequence h, in which people crossed paths and occluded one another, saw a slight drop in performance compared to the other sequences, but we were still able to maintain 81.3% purity (other sequences ranged from 80.5% to 98.3%). These numbers indicate that our model was mostly successful in maintaining personal identity through occlusion, as seen in Fig. 6.11

Finally, for spatial fitting, the bounding boxes generally fit the ground truths tightly, as evidenced by Figures 6.7, 6.8c and 6.8d. Both the body and head had a mean fit of 0.87 (1 being optimal). As seen in Figure 6.7, the fit often suffered from partially visible bodies and heads that occurred when people entered and exited the scene.

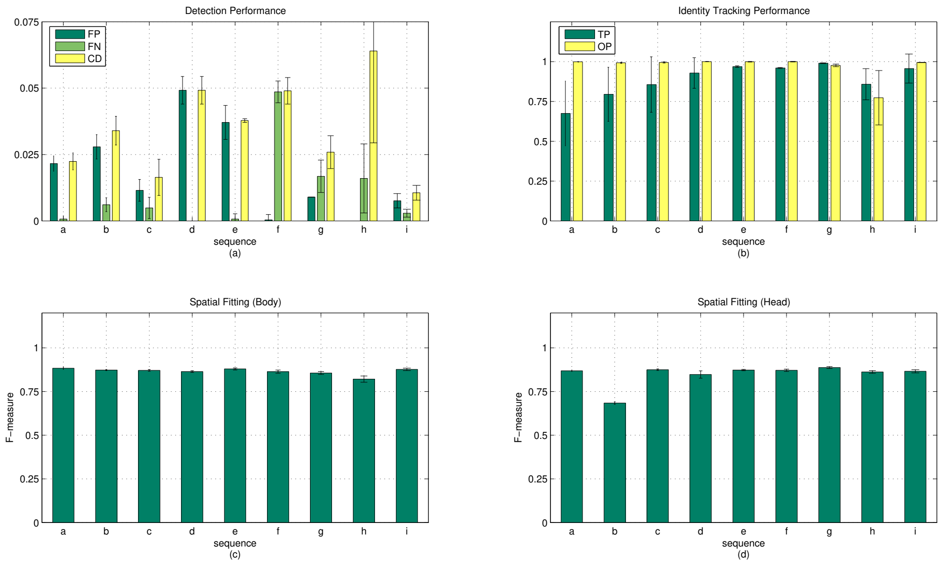


Figure 6.8: *Multi-Person Head and Body Tracking Results*. The *detection performance* plot in *(a)* measures the ability of the model to estimate the correct number and placement of people in the scene. Measures shown include the normalized *false positive* ($\overline{FP}$) and *false negative* ($\overline{FN}$) error rates (per person, per frame), and the *counting distance* ($\overline{CD}$) (near-zero values are good, see text). The *tracking performance* plot in *(b)* measures the ability of the model to persistently track people over time. *Tracker purity* ($\overline{TP}$) and *object purity* ($\overline{OP}$) measure the consistency with which the ground truths and estimates were properly identified. TP and OP values near 1 indicate good performance. Plots *(c)* and *(d)* show the *spatial fitting* results (how well the tracker bounding boxes fit the ground truth) for the body and the head over the nine sequences. Overlap between the estimate and ground truth are measured using the F-measure. A value of 1 indicates a perfect fit, a value of zero indicates no overlap. In each plot, the standard deviation is represented by error bars (cases where no error bar is visible indicates $std = 0$).

6.5.2 Advertisement application performance

To evaluate the performance of the advertisement application, the results from our model were compared with a ground truth where the WVFOA was labeled for each person as either *focused* or *unfocused*. In our evaluation, we considered the following criteria: (1) the number of people exposed to the advertisement, (2) the number of people who looked, or *focused*, at the advertisement, (3) the number of events where someone *focused* on the advertisement (look-events), and (4) the frame-based and (5) event-based recognition rates of the WVFOA. Results for the ad application evaluation appear in Figure 6.9.

Regarding criterion 1, over the entire test set, 22 people passed the advertisement, while our model estimated a value of 22.15 (average for all runs, standard dev. $=$ .17), in Figure 6.9a we can see that the number of people was correctly estimated for all sequences except a, c, and h.

With respect to criterion 2, from a total of 22 people, 20 actually *focused* on the advertisement. Our model estimated a value of 20.75 (standard dev. $=$ .09). Figure 6.9b shows perfect results for all sequences except a, d, and h.

For criterion 3, we defined a look-event as a *focused* state for a continuous period of time of 3 frames or more. The total number of look-events in the test data set was 22, 21 of which our system recognized on average (standard dev. = .89). This result was determined through a standard symbol matching technique (see below). However, our model estimated 37 total look-events on average (standard dev. $= 1.1$). This disparity can be partially attributed to problems in head-pose estimation for heads partially outside the image as people enter or leave. The look-event estimation results would improve if we did not consider WVFOA in these cases. Also, the look event duration of 3 frames is quite strict, and some erroneous looks were generated by noise.

Finally, to evaluate the overall quality of WVFOA estimation, we compute recognition rates for event-based WVFOA and frame-based WVFOA using the aforementioned *F-measure*. To compute the event-based F, the ground truth and estimated WVFOA are segmented over the entire sequence into focused and unfocused events, symbol matching is performed accounting for temporal alignment, and F is computed on matched segments. Results are shown in Figure 6.9d. The overall event-based F is 0.76 (standard dev. = .13). The frame-based F is computed by matching the estimated WVFOA for each frame to the ground truth. The overall frame-based F-measure is 0.76 (standard dev. $= .06$). Poor frame-based results in sequence g occurred because the subject *focused* for a very short time as he entered the field of view (0.3s), during which time his head was only partially visible. However, our model still managed to detect at the *event* level with $F = .75$.

6.5.3 Varying the number of particles

To study the model's dependency on the number of samples, we conducted experiments on sequence i (the most complex in terms of number of simultaneous people), varying the number of samples $N = \{50, 100, 200, 600, 800, 1000\}$. The results are shown in Fig. 6.10. For all N, the model correctly estimated the number of people who passed and the number of people who looked. With less samples, the spatial fitting and detection (as measured by $1 - \overline{CD}$) suffered. The head tracking and head-pose estimation was noticeably shakier with less samples, and the WVFOA estimation suffered as a consequence. This is shown by the increased error in the number of estimated looks for low sample counts. The model stabilized around approximately $N = 600$.

6.6 Conclusion

In this chapter, we have introduced the WVFOA problem and presented a principled probabilistic approach to solving it. Our work thus contributes in terms of both problem definition and statistical vision modeling. Our approach expands on state-of-the-art RJMCMC tracking models, with novel contributions to object modeling, likelihood modeling, and the sampling scheme. We applied our model to a real-world application and provided a rigorous objective evaluation of its performance. From these results we have shown that our proposed model is able to track a varying number of moving people and determine their WVFOA with good quality. Our model is general and can easily be adapted to other similar applications.

For future work, investigating the usefulness of using a spatially dependent face/pose detector as an additional feature is one possible avenue. Other work might include modeling multiple human-to-human interaction using WVFOA.

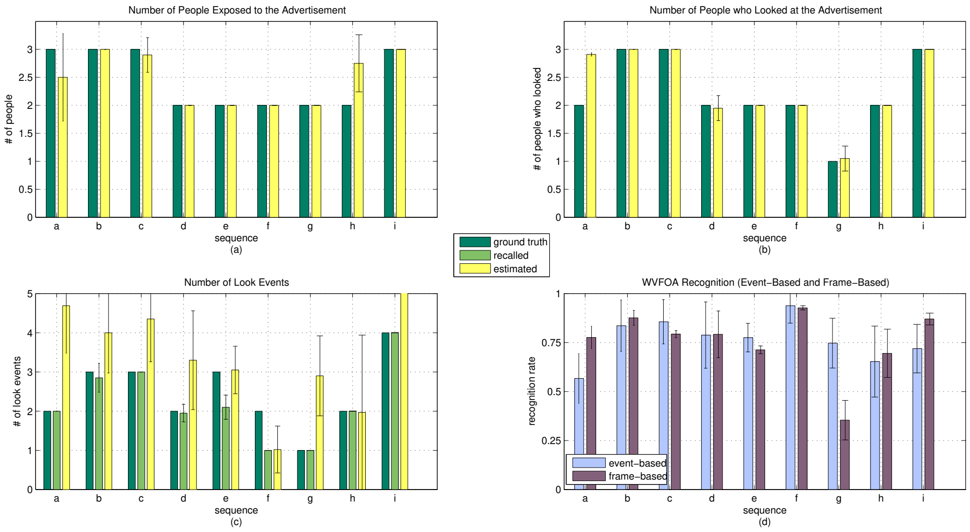


Figure 6.9: *Ad Application Results*. The first three plots show *(a)* the number of people exposed to the advertisement, *(b)* the number of people who looked at the advertisement, and *(c)* the number of "look events" for the nine sequences. The dark green bars represent the ground truth, while the yellow bars represent our model's estimates. In *(c)*, the light green bars represent the number of actual look events detected by our system. In each plot, the standard deviation is represented by error bars (cases where no error bar is visible indicates $std = 0$). Plot *(d)* shows the overall recognition rate of *focused* and *unfocused* states (calculated based on events and based on frame counts).

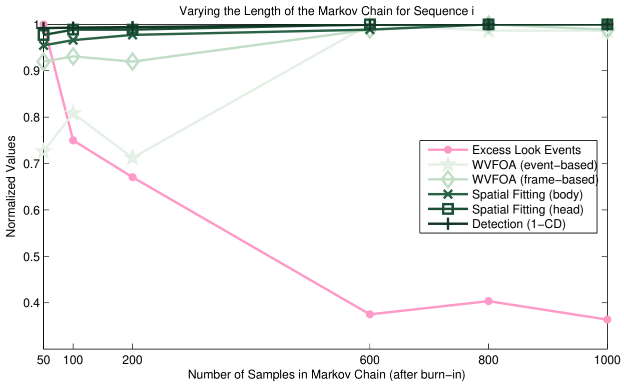


Figure 6.10: *Varying the Number of Samples in the Markov Chain*. As more samples used, various performance gauges increase, as seen here for sequence i. Excess (false alarm) look events (pink) detected by the system drop as more samples are added, the WVFOA recognition improves (both for event and frame based), spatial fitting for the body and head improves, and *detection* performance increases (as measured by $1 - \overline{CD}$). Note that the measures have been normalized to appear on the same axis.

Figure 6.11: *Tracking and WVFOA Results.* Results for four sequences, b, e, h, and i. A summary plot of the WVFOA performance is provided in the last pane of each row. For details on interpreting the plots and symbols, refer to Fig 6.7. Here, we can see the WVFOA performance was nearly perfect for sequence b and exhibited slight errors in sequence i. The 2^{nd} person (green) in sequence e suffered from prematurely estimating a *focused* state. Sequence h suffered some ambiguities due to the loss of head tracking as people crossed paths. The last frame of sequence i shows a FP error generated as a tracker was placed where no ground truth was present (though the subject is half visible as he exits the scene). Such situations can cause ambiguity problems.

Chapter 7

Conclusions and future works

7.1 Conclusions

This thesis was conducted in the context of the Augmented Multi-party Interaction (AMI) project which targets computer enhanced multi-modal interaction in the contexts of meetings. This thesis was also conducted through the Interactive Multi-modal Information Management (IM2) Swiss NCCR [1] within the Multiple Camera Tracking and Activity Recognition (MUCATAR) Project. The MUCATAR Project was aimed at designing probabilistic based models and algorithms for the simultaneous tracking of people and recognition of their activities. More precisely, the goal of this thesis was to study the recognition of people's visual focus of attention (VFOA) from video recordings, assuming low to medium resolution imagery. No electronic sensors were to be used to solve the task. Also, the image resolution did not allows for the use of 3D models, or other models such as active appearance models (AAM) requiring high resolution imagery. Thus, we adopted an approach relying on the tracking of the head location and pose, and then on the VFOA estimation from the head pose (and position if necessary). The contributions of this thesis are the following.

Head pose video database:
We built a publicly available head pose video database using a magnetic field head location and orientation tracker. This database is an important contribution of this thesis. It was built due to the absence in the tracking community of an evaluation database. Although, few researchers who used head pose video data with pose ground truth for evaluation, their database were private, thus did not allow algorithms evaluation and comparison. Furthermore, most of the time, these databases were constituted by very short recordings.

Head pose tracking:
We proposed a probabilistic framework for joint head tracking and pose estimation. It relies on a discretization of the head pose space and the learning of appearance models from various cues (texture, skin color, background subtraction). Different tracking methods were formulated within a Bayesian formalism solved through sampling techniques. From a rigorous evaluation of the head pose tracking methodologies, we showed that jointly tracking the head location and pose is more efficient than tracking the head location, and then estimating the pose. Secondly, we showed that the Rao-Blackwellized

[1] National Center for Competence in Research

version of the mixed state particle filter was more efficient than the raw MSPF or the MCMC that we studied. Finally, we also showed that the use of a data driven component allowed for more reliable tracking performances.

VFOA recognition in meetings:
We generalized to more complex situations previous work about modeling the VFOA of static persons from their head pose. In our work, due to the physical setup, the whole range of head pose variation was required (pan and tilt). Furthermore, the set of potential VFOA targets to be recognized was larger. In our setup, there were six potential VFOA targets while in previous work, there were only 3 potential VFOA. We modeled the VFOA using classical Gaussian mixture models (GMM), and hidden Markov models (HMM). The hidden state of these models were the potential VFOA targets. The head poses, obtained either from the RBPF head pose tracker or the head pose ground truth (GT) from the evaluation database, were used as observations. Two methods were used to learn the observation models. The first method was based on using a training approach from annotated data, the second method was based on a geometric approach using the 3D geometry of the meeting room. Also, because people have personal ways to look at targets, unsupervised maximum a posteriori adaptation (MAP) of the generic VFOA models is applied to the input test data. From a thorough evaluation of our models, we showed that there can be a significant degradation when passing from clean GT data to head pose tracking data that can be sometimes noisy due to short time tracking failure. Secondly, we showed that unsupervised MAP adaptation can significantly improve the VFOA recognition. Thirdly, while the geometric approach achieves similar results than the training-based approach, when using the head pose GT data, it generates slightly better results when using the tracking head pose estimates. One of the reason is that, the geometric model parameters are noise free in the sense that they do not depend on noisy head pose training data estimates. Keeping in mind that VFOA annotation is a very tedious task, the geometric model is very promising for future research. It will allow us to study the VFOA of moving people where generating training data to apply the training based approach is infeasible.

Moving people VFOA recognition:
As a preliminary work about the study of VFOA of moving people, we proposed to solve the following task: given an advertisement posted on a glass windows, track the people passing by the window and recognize wether or not they are focusing their attention on the advertisement. We termed this task wandering VFOA (WVFOA) recognition. Solving this task requires the tracking of the body location, head location and pose of multiple people, and building a VFOA model allowing to handle moving people. We used reversible jump Markov Chain Monte Carlo (RJ-MCMC) to achieve the multi-person configuration tracking. In a second step, head location and pose output of the tracker are used in a GMM framework to recognize whether or not people are focusing on the advertisement. The evaluation of the method we proposed showed that, though the dimension of the multi-person state space was very high, 10 dimension for each person present in the scene, the tracking method was very reliable. The WVFOA model was also very efficient to recognize people's VFOA. The WVFOA, including the geometric VFOA modeling, is a good starting point to study the VFOA of moving people when there are multiple possibly moving VFOA targets.

7.2 Limitations and future work

In this thesis, although we deeply investigated probabilistic head pose tracking and VFOA modeling, further investigations can be conducted. In the following we analyze some of the limitations of our work and propose future research directions, first about head pose tracking, then about VFOA modeling.

7.2.1 Head pose tracking

Appearance model adaptation:
Our head pose tracking algorithms are based on appearance models, built from texture observations, skin color observation, and binary feature from background subtraction. In this work, only the feature extraction process has been made adaptive. The skin color distribution and the background model were temporally adapted to the new data. However, the appearance models were not adapted to the individuals. From our experience, better tracking performance were achieved with people who were more similar to the appearance models. Thus, developing methods able to be adapted to individual people would be quite beneficial. Investigations can be made towards using global 3D head models, where adaptation is more natural than with our discretized setting. However, this will require more robust model fitting techniques that can be applied to low or medium head image resolution.

Multi-sensor tracking:
In this work, we studied the VFOA recognition using only a single camera view of people. In the context of the meeting room, it is worth noticing that there are multiple camera views. A person is always recorded from at least two camera views. Our tracking framework could be generalized to a multi-view probabilistic tracking framework. We have noticed that the estimation of near frontal pose are more reliable than near profile pose estimation. If cameras recording from orthogonal views are available, complementary information are provided by the cameras. Prior on the reliability of each of the camera views depending on the pose estimated by a camera could be used in a fusion process. Also, investigations can be made to integrate in our tracking model speech signals from microphone arrays to allow tracking initialization or re-initialization after tracking failures due to occlusions. Speech can also be used to improve the proposal function when generating candidate head locations.

7.2.2 VFOA recognition

Multiple moving people VFOA recognition:
In this thesis, we have studied thoroughly the VFOA recognition of static persons in a meeting context and the VFOA of moving person potentially looking at one static VFOA target. Interesting future investigations would be to study the VFOA of moving persons with possibly moving VFOA targets. This task is very challenging as in this case, the class conditional distributions for the VFOA targets need to be dynamic. The geometric VFOA model that we proposed in this thesis is a good starting point to study the VFOA of moving people. It can be a more straightforward approach to build class conditional distributions for any kind of possible person-VFOA target configurations, while in a static off line approach using training data, generating training data for all the possible configurations becomes quickly

infeasible.

Multi-sensor VFOA recognition:
VFOA recognition, as head pose tracking, can make use of the multiple cameras and microphones available in meeting rooms. In conversations, VFOA is tightly connected to speech structure, as people usually gaze at the speaker. Thus speech features can bring complementary information to the visual gaze information. Because of the relation between VFOA and conversations structures, investigations should be conducted about using VFOA as a features to estimates conversation patterns such as finding a person addressees, or dominant people in a meeting.

We can also considerer the focus of a person as being a complementary information to the focus of another person. The reason is that, in groups, people's gazes are highly correlated. People mutually gaze at each other or gaze to the same target. Thus, investigating joint modeling of the VFOA of all the persons instead of considering people's VFOA as independent is natural.

Bibliography

[1] S.R.H. Langton, R.J. Watt, and V. Bruce, "Do the Eyes Have it ? Cues to the Direction of Social Attention," *Trends in Cognitive Sciences*, vol. 4(2), pp. 50–58, 2000.

[2] M. J. Jones and J. M. Rehg, "Statistical Color Models with Application to Skin Detection," *International Journal of Computer Vision*, , no. 1, pp. 81–96, 2002.

[3] M. Yang and N. Ahuja, "Gaussian Mixture Model for Human Skin Color and Its Application in Image and Video Databases," in *Conference on Storage and Retrieval for Image and Video Databases (SPIE 99)*, 1999, pp. 458–466.

[4] S.L. Phung, A. Bouzerdoum, and D. Chai, "Skin Segmentation Using Color Pixel Classification: Analysis and Comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 148–154, 2005.

[5] J. Yang, L. Weier, and A. Waibel, "Skin-Color Modeling and Adaptation," in *Asian Conference On computer Vision (ACCV)*, 1998, pp. 687–694.

[6] M. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms," *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 19, no. 12, pp. 1338–1359, 1997.

[7] S.G. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.

[8] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative Study of Texture Measures with Classification based on Featured Distribution," *Pattern Recognition Letters*, vol. 29, no. 1, pp. 51–59, 1996.

[9] D. Lowe, "Distinctive Image Features from Scale Invariant Keypoints," *International Journal of Computer Vision*, vol. 20, pp. 91–110, 2003.

[10] M. Black and Y. Yacoob, "Recognizing Facial Expressions in Image Sequences using Local Parameterized Models of Image Motion," *International Conference on Computer Vision (ICCV)*, 1995.

[11] M.J. Jones and T. Poggio, "Multidimensional Morphable Models," in *International Conference on Computer Vision (ICCV)*, 1998, pp. 683–688.

[12] S. Niyogi and W. Freeman, "Example-based Head Tracking," in *International Conference on Automatic Face and Gesture Recognition (AFGR)*, 1996, pp. 374–378.

[13] M. Isard and A. Blake, "Condensation: CONDENSATION Density Propagation for Visual Tracking," *International Journal of Computer Vision*, pp. 5–28, 1998.

[14] D. Gatica-Perez, G. Lathoud, I. McCowan, J.-M. Odobez, and D. Moore, "Audio-Visual Speaker Tracking with Importance Particle Filters," in *IEEE International Conference on Image Processing (ICIP)*, 2003.

[15] J.-M. Odobez, D. Gatica-Perez, and S.O. Ba, "Embedding Motion in Model-Based Stochastic Tracking," *IEEE Transaction on Image Processing*, vol. 15, no. 11, pp. 3514–3530, 2005.

[16] A. Blake and M. Isard, *Active Contours*, Springer-Verlag London, 1998.

[17] L. Wiskott, J.-M. Fellous, N. Kruger, and C. Von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *Intelligent Biometric Techniques in Fingerprints and Face Recognition*, pp. 355–396, 1999.

[18] T. F. Cootes and C.J. Taylor, "Active Shape Models-Their Training and Application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[19] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[20] S. Srinivasan and K. L. Boyer, "Head Pose Estimation Using View Based Eigenspaces," in *International Conference on Pattern Recognition (ICPR)*, 2002.

[21] S. Nayar, H. Murase, and S. Nene, "Parametric Appearance Representation," *Early Visual Learning, Oxford University Press*, 1996.

[22] S. McKenna and S. Gong, "Real Time Face Pose Estimation," *Real-Time Imaging*, vol. 4, no. 5, pp. 333–347, 1998.

[23] V. Kruger, S. Bruns, and G. Sommer, "Efficient Head Pose Estimation with Gabor Wavelet Networks," in *British Machine Vision Conference (BMVC)*, 2000.

[24] S Roweis and L. Sau, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, 2000.

[25] B. Raytchev, I. Yoda, and K Sakaue, "Head Pose Estimation by Nonlinear Manifold Learning," in *International Conference on Pattern Recognition (ICPR)*, 2004, vol. 4, pp. 462 – 466.

[26] Y. Fu and T. S. Huang, "Graph Embedded Analysis for Head Pose Estimation," in *International Conference on Automatic Face and Gesture Recognition (AFGR)*, 2006, pp. 3–8.

[27] H. Rowley, .S Baluja, and T. Kanade, "Neural Network-Based Face Detection," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996.

[28] D. Valentin, H. Abdi, and A. O'Toole, "Principal Component and Neural Network Analysis of Face Images: Explorations Into the Nature of Information Available for Classifying Faces by Gender," *In C. Dowling, F.S. Roberts, P. Theuns (Eds.) Progress in Mathematical Psychology. Hillsdale: Lawrence Erlbaum.*, 1996.

[29] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: an Application to Face Detection.," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997.

[30] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, vol. 1, pp. 511–518.

[31] Schapire R. E., Freund Y., P. Bartlett, and W. S. Lee, "Boosting the Margin: a New Explanation for the Effectiveness of Voting Methods," in *International Conference on Machine Learning (IMCL)*, 1997, pp. 322–330.

[32] P. Viola and M.J. Jones, "Fast Multi-view Face Detection," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[33] G. Heusch, Y. Rodriguez, and S. Marcel, "Local Binary Patterns as an Image Preprocessing for Face Authentication," in *International Conference on Automatic Face and Gesture Recognition (AFGR)*, 2006.

[34] L. Zhao, G. Pingali, and I. Carlbom, "Real-Time Head Orientation Estimation Using Neural Networks," in *International Conference on Image Processing (ICIP)*, 2002, pp. 297–300.

[35] N. Gourier, J. Maisonnasse, D. Hall, and J.-L. Crowley, "Head Pose Estimation on Low Resolution Images," in *CLEAR Evaluation Workshop*, 2006.

[36] R. Rae and H. Ritter, "Recognition of Human Head Orientation Based on Artificial Neural Networks," *IEEE Transaction on Neural Network*, vol. 9(2), pp. 257–265, 1998.

[37] Y. Li, S. Gong, J. Sherrah, and H. Liddell, "Support Vector Machines based Multi-view Face Detection and Recognition," *Image and Vision Computing*, vol. 22, pp. 413–427, 2004.

[38] P. Wang and Q. Ji, "Multi-View Face Tracking with Factorial and Switching HMM," *Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05)*, vol. 1, pp. 401–406, 2005.

[39] H. Schneiderman and H. Kanade, "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 45–51.

[40] K. K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[41] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 19, no. 7, pp. 696–710, 1997.

[42] C. Kervrann, F. Davoine, and P. Perez, "Generalized Likelihood Ratio-based Face Detection and Extraction of Mouth Features," in *Int. Conf. on Audio and Video-Based Biometric Person Authentication, AVBPA'97*, 1997.

[43] H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 151–177, 2002.

[44] Y. Wu and K. Toyama, "Wide Range Illumination Insensitive Head Orientation Estimation," in *International Conference on Automatic Face and Gesture Recognition (AFGR)*, 2001.

[45] L. M. Brown and Y.-L. Tian, "A Study of Coarse Head Pose Estimation," in *IEEE Workshop on Motion and Video Computing*, 2002.

[46] P. J. Phillips, H. Moon, and P. J. Rauss, "The FERET Evaluation Methodology for Face Recognition Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 10, pp. 1090–1104, 2000.

[47] Sim T., S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, no. 12, pp. 615 – 1618, 2003.

[48] N. Gourier, D. Hall, and J. L. Crowley, "Estimating Face Orientation from Robust Detection of Salient Facial Features," in *Pointing 2004, ICPR international Workshop on Visual Observation of Deictic Gestures*, 2004, pp. 183–191.

[49] G.J Edwards, C.J. Taylor, and T.F. Cootes, "Learning to Identify and Track Faces in Image Sequences," in *International Conference on Computer Vision (ICCV)*, 1997, pp. 317–322.

[50] G.J. Edwards, T.F. Cootes, and C.J. Taylor, "Face Recognition Using Active Appearance Models," in *European Conference on Computer Vision (ECCV)*, 1998, pp. 581 – 595.

[51] D. Dornaika and J. Ahlberg, "Fast and Reliable Active Appearance Model Search for 3-D Face Tracking," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 4, pp. 1838–1853, 2004.

[52] S. Hamlaoui and F. Davoine, "Facial Action Tracking using Particle Filters and Active Appearance Models," in *Joint Conference on Smart Objects and Ambient Antelligence: Annovative Context-Aware services: Usages and Technologies*, 2005, pp. 165 – 169.

[53] I. Matthews and S. Baker, "Active Appearance Models Revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135 – 164., 2004.

[54] C. M. Christoudias, L. P. Morency, and T. Darrell, "Light Field Appearance Manifolds," in *European Conference on Computer Vision (ECCV)*, 2004, pp. 481–493.

[55] J. Ahlberg, "CANDIDE-3, An Updated Parameterized Face," Tech. Rep. liTH-ISY-R-2326, Dept. of Electrical Engineering, Linkoping University, 2001.

[56] M. Rydfalk, "CANDIDE, a Parameterized Face," Tech. Rep. LiTH-ISY-I-866, Dept. of Electrical Engineering, Linkoping University, 1987.

[57] I.A. Essa and A. Pentland, "A vision System for Observing and Extracting Facial Action Parameters," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 76–83.

[58] R. Zhang and Z. Zhang, "Model-based Head Pose Tracking with Stereo-Vision," Tech. Rep. MSR-TR-2001-102, Microsoft Research, 2001.

[59] M. Brand and R. Bothika, "Flexible Flow for 3D Nonrigid Tracking and Shape Recovery," Tech. Rep. TR-2001-38, Microsoft Research, 2001.

[60] Y.-J. Chang and Y.-C. Chen, "Facial Model adaptation from a Monocular Image Sequence using a Textured Polygonal Model," *Signal Processing: Image Communication*, , no. 17, pp. 373–392, 2002.

[61] M. Malciu and F. Preteux, "A Robust Model-based Approach for 3D Head Tracking in Video Sequences," in *International Conference on Automatic Face and Gesture Recognition (AFGR)*, 2000, pp. 169–174.

[62] M. Yokoyama and T. Poggio, "A Contour based Moving Object Detection and Tracking," *International Conference on Computer Vision (ICCV)*, 1995.

[63] Y. Mae, Y. Shirai, and J. Miura, "Object Tracking in Cluttered Background Based on Optical Flow and Edges," in *International Conference on Pattern Recognition (ICPR)*, 1996, vol. 1, pp. 196–200.

[64] E. Memin and P. Perez, "Dense Estimation and Object-Based Segmentation of the Optical Flow with Robust Techniques," *IEEE Transaction on Image Processing*, vol. 7, no. 5, pp. 703–719, 1998.

[65] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt, "Performance of Optical Flow Techniques," in *International Conference on Pattern Recognition (ICPR)*, 1992, pp. 236–242.

[66] D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications," in *International Conference Computer Vision (ICCV)*, 1999, pp. 1197–1203.

[67] A. Bhattacharyya, "On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions," *Bulletin of Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.

[68] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, pp. 35–45, 1960.

[69] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, pp. 174–188, 2002.

[70] S Julier and J. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *International Symposium on Aerospace/Defense Sensing, Simul. and Controls*, 1997.

[71] E. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *IEEE Symposium 2000 (AS-SPCC)*, 2000.

[72] S. J. Julier, "The Scaled Unscented Transformation," in *American Control Conference*, 2002.

[73] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel Approach to Nonlinear and Non-Gaussian Bayesian State Estimation," *Radar and Signal Processing, IEE Proceedings F*, pp. 107–113, 1993.

[74] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, 2000.

[75] D. Crisan and A. Doucet, "A Survey of Convergence Results on Particle Filtering for Practitioners," *IEEE Transaction on Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002.

[76] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Models," *Journal of Computational and Graphical Statistics*, pp. 1–25, 1996.

[77] P. Perez, J. Vermaak, and A. Blake, "Data Fusion for Visual Tracking with Particles," *Proceedings of the IEEE: Issue on State Estimation*, vol. 93, no. 3, pp. 495– 513, 2004.

[78] D. Gatica-Perez, G. Lathoud, I. McCowan, J.-M. Odobez, and M. Moore, "Audio-Visual Speaker Tracking with Importance Particle Filters," in *International Conference on Image Processing (ICIP)*, 2004.

[79] M. Isard and B. Blake, "ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework," *Lecture Notes in Computer Science*, vol. 1406, pp. 893–908, 1998.

[80] M.K. Pitt and N. Shephard, "Filtering via Simulation: Auxiliary Particle Filters," *Journal of the American Statistical Association*, vol. 94, pp. 446–590, 1999.

[81] C.H. Morimoto and M.R. Mimica, "Eye Gaze Tracking Techniques for Interactive Applications," *Computer Vision and Image Understanding*, vol. 98, pp. 4–24, 2005.

[82] J. Babcock and J. Pezl, "Building a Lightweight Eye Tracking Headgear," in *ACM SIGCHI: Eye Tracking Research and Applications*, 2004, pp. 109–114.

[83] J. Pelz, R. Canosa, J. Babcock, D. Kucharczyk, A. Silver, and D. Konno, "Portable Eyetracking: A Study of Natural Eye Movements," in *Human Vision and Electronic Imaging*, 2000.

[84] T. Ohno and N. Mukawa, "A Free-head, Simple Calibration, Gaze Tracking System that Enables Gaze-based Interaction," in *Symp. on Eye tracking Research and Applications*, 2004, pp. 115 – 122.

[85] A. E Kaufman, A. Bandopadhay, and B. D. Shaviv, "An Eye Tracking Computer User Interface," in *Research Frontier in Virtual Reality Workshop*, 1993, pp. 78–84.

[86] D.A. Robinson, "A Method of Measuring Eye Movements Using a Scleral Search Coil in a Magnetic Field," *IEEE Transaction on Biomedical Engeneering*, vol. 10, pp. 137–15, 1963.

[87] J. M. Henderson, "Human Gaze Control During Real-world Scene Perception," *Trends in Cognitive Sciences*, vol. 7, no. 11, pp. 498–504, 2003.

[88] D. J. Parkhurst and E. Niebur, "Scene Content Selection by Active Vision," *Spatial Vision*, vol. 16, no. 2, pp. 125–154, 2003.

[89] G. Underwood, P. Chapman, N. Broklehurst, J. Underwood, and D. Crundall, "Visual Attention While Driving: Sequences of Eye Fixation made by Experienced and Novice Drivers," *Ergonomics*, vol. 46, pp. 629–646, 2003.

[90] M. Hayhoe, "Visual Memory and Motor Planning in a Natural Task," *Journal of Vision*, , no. 3, pp. 49–63, 2003.

[91] M. K. Tanenhaus, M. J. Spivey-Knowlton, K. M. Eberhard, and J. Sedivy, "Integration of Visual and Linguistic Information in Spoken Language Comprehension," *Science*, vol. 268, no. 5217, pp. 1632–1636, 1995.

[92] R. Pieters and M. Wedel, "Attention Capture and Transfer in Advertising: Brand, Pictorial and Text Size Effects," *Journal of Marketing*, vol. 68, pp. 36–50, 2004.

[93] R. Pieters and L. Warlop, "Visual Attention during Brand Choice: The Effect of Time Pressure and Task Motivation," *International Journal of Research Marketing*, 1998.

[94] F. A. Proudlock, H. Shekhar, and I. Gottlob, "Coordination of Eye and Head Movements during Reading," *Investigative Ophtalmology & Visual Science*, pp. 2991–2998, 2003.

[95] J.E. McGrath, *Groups: Interaction and Performance*, Prentice-Hall, 1984.

[96] M. Argyle and D. Jean, "Eye Contact, Distance and Affiliation," *Sociometry*, pp. 289–304, 1965.

[97] R. Vertergaal, R. Slagter, G.C Van der Veer, and A. Nijholt, "A Gaze Pattern in Conversations: There is More to Conversational Agents Than Meets the Eyes," in *Conference on Human Factors in Computing Systems*, 2001, pp. 301–308.

[98] N. Jovanovic and H.J.A. Op den Akker, "Towards Automatic Addressee Identification in Multi-Party Dialogues," in *5th SIGDIAL Workshop onDiscourse and Dialogue*, 2004.

[99] R. Vertergaal, G.C. Van Der Veer, and H. Vons, "Effects of Gaze on Multiparty Mediated Communication," in *Graphics Interface*, 2000, pp. 95–102.

[100] S. Duncan-Jr, "Some Signals and Rules for Taking Speaking Turns in Conversations," *Journal of Personality and Social Psychology*, vol. 23(2), pp. 283–292, 1972.

[101] D. Novick, B. Hansen, and K. Ward, “Coordinating Turn Taking with Gaze,” in *International Conference on Spoken Language Processing (ICSLP)*, 1996.

[102] R. Stiefelhagen, J. Yang, and A. Waibel, “Modeling Focus of Attention for Meeting Indexing based on Multiple Cues,” *IEEE Transactions on Neural Networks*, vol. 13(4), pp. 928–938, 2002.

[103] CLEAR Evaluation, “CLEAR Evaluation and Workshop,” 2006.

[104] Ascencion Technology, “Flock of Birds,” .

[105] M. W. Spong and M. Vidayasagar, *Robot Dynamics and Controls*, John Wiley and Sons Inc., 1998.

[106] Z. Zhang, “A Flexible New Technique for Camera Calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 11, pp. 1330–1334, 2000.

[107] K. Toyama and A. Blake, “Probabilistic Tracking in metric Space,” in *International Conference on Computer Vision (ICCV)*, 2001.

[108] M. Voit, K. Nickel, and R. R. Stiefelhagen, “Neural Network based Head Pose Estimation and Multi-view Fusion,” in *CLEAR Evaluation Campaign and Workshop*, 2006.

[109] J. Tu, Y. Fu, Y. Hu, and T. Huang, “Evaluation of Head Pose Estimation for Studio Data,” in *CLEAR Evaluation Campaign and Workshop*, 2006.

[110] N. Gourier, J. Maisonnasse, D. Hall, and J.L. Crowley, “Head Pose Estimation on Low Resolution Images,” in *CLEAR Evaluation Campaign and Workshop*, 2006.

[111] Z. Zhang, Y. Hu, M. Liu, and T. Huang, “Head Pose Estimation in Seminar Room Using Multi View Face Detectors,” in *CLEAR Evaluation Campaign and Workshop*, 2006.

[112] J. Vermaak and A. Blake, “Nonlinear Filtering for Speaker Tracking in Noisy and Reverberant Environments,” in *International Conf. on Acoustic Speech Signal Processing (ICASSP)*, 2000.

[113] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Readings in Speech Recognition*, vol. 53A(3), pp. 267–296, 1990.

[114] K. Otsuka, Y. Takemae, J. Yamato, and H. Murase, “A Probabilistic Inference of Multiparty-Conversation Structure based on Markov-Switching Models of Gaze Patterns, Head directions, and Utterances,” in *International Conf. on Multimodal Interface (ICMI)*, 2005, pp. 191–198.

[115] E. G. Freedman and D. L. Sparks, “Eye-Head Coordination During Head-Unrestrained Gaze Shifts in Rhesus Monkeys,” *Journal of Neurophysiology*, vol. 77, pp. 2328–2348, 1997.

[116] I.V. Malinov, J. Epelboim, A.N. Herst, and R.M. Steinman, “Characteristics of Saccades and Vergence in two Kinds of Sequential Looking Tasks,” *Vision Research*, 2000.

[117] P. Smith, M. Shah, and N. Da Vitoria Lobo, “Determining Driver Visual Attention with One Camera,” *IEEE Transaction on Intelligent Transportation Systems*, vol. 4.(4), pp. 205–218, 2004.

[118] Y. Matsumoto, T. Ogasawara, and A. Zelinsky, "Behavior Recognition based on Head Pose and Gaze Direction Measurement," in *Conference on Intelligent Robots and Systems*, 2002.

[119] N.M. Robertson and I.D. Reid, "Estimating Gaze Direction from Low-Resolution Faces in Video," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 402–415.

[120] M. Danninger, R. Vertegaal, D.P. Siewiorek, and A. Mamuji, "Using Social Geometry to Manage Interruptions and Co-worker Attention in Office Environments," in *Conference on Graphics Interfaces*, 2005.

[121] M. Hayhoe and D. Ballard, "Eye Movements in Natural Behavior," *Trends in Cognitive Sciences*, vol. 9(4), pp. 188–194, 2005.

[122] S. Baron-Cohen, "How to Build a Baby that Can Read Minds: Cognitive Mechanisms in Mindreading," *Cahier de Psychologies Cognitive*, vol. 13, pp. 513–552, 1994.

[123] J.-M. Odobez, "Focus of Attention Coding Guidelines," Tech. Rep. IDIAP-COM-2, IDIAP Reasearch Institute, 2006.

[124] J.L. Gauvain and C. H. Lee, "Bayesian Learning for Hidden Markov Model with Gaussian Mixture State Observation Densities," *Speech Communication*, vol. 11, pp. 205–213, 1992.

[125] R. Stiefelhagen, *Tracking and Modeling Focus of Attention*, Ph.D. thesis, University of Karlsruhe, 2002.

[126] K. Smith, D. Gatica-Perez, and J.M. Odobez, "Using Particles to Track Varying Numbers of Objects," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, June 2005.

[127] V. Starr and C.A. Lowe, "The Influence of Program Context and Order of ad Presentation on Immediate and Delayed Responses to Television ads," *Advances in Consumer Research*, vol. 22, pp. 184–190, 1995.

[128] J. MacCormick and A. Blake, "A Probabilistic Exclusion Principle for Tracking Multiple Objects," in *International Conference on Computer Vision (ICCV)*, Kerkyra, Greece, Sep. 1999.

[129] H. Tao, H. Sawhney, and R. Kumar, "A Sampling Algorithm for Detection and Tracking Multiple Objects," in *ICCV Workshop on Vision Algorithms*, Kerkyra, Sept. 1999.

[130] M. Isard and J. MacCormick, "BRAMBLE: a Bayesian Multi-Blob Tracker," in *International Conference on Computer Vision (ICCV)*, Vancouver, Jul. 2001.

[131] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe, "A Boosted Particle Filter: Multi-Target Detection and Tracking," in *European conference on Computer vision (ECCV)*, Prague, May 2004.

[132] T. Yu and Y. Wu, "Collaborative Tracking of Multiple Targets," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington DC, June 2004.

[133] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based Particle Filter for Tracking Multiple Interacting Targets," in *European conference on computer vision (ECCV)*, Prague, May 2004.

[134] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Crowded Environment," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington DC, June 2004.

[135] C. Stauffer and E. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, Ft. Collins, CO, June 1999.

[136] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan, "An Introduction to MCMC for Machine Learning," *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[137] P. Green, "Reversible Jump MCMC Computation and Bayesian Model Determination," *Biometrika*, vol. 82, pp. 711–732, 1995.

[138] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, pp. 1805–1819, 2005.

[139] K. Smith, D. Gatica-Perez, S. Ba, and J.M. Odobez, "Evaluating Multi-Object Tracking," in *CVPR Workshop on Empirical Evaluation Methods in Computer Vision*, San Diego, June 2005.

Curriculum Vitae

Silèye Oumar Ba
address: IDIAP Research Institute
04 Rue Du Simplon , 1920, Martigny, Switzerland
tel: +41277217761
emails: sba@idiap.ch, sileyeoba@yahoo.fr

Research

October 2002 to March 2007: Research Assistant at IDIAP Research Institute working probabilistic methods for head tracking and pose estimation, and for visual focus of attention modeling and recognition.

From April 2002 to September 2002: Intern at Institut National de Recherche en Agronomie (INRA), in Jouy en Josas Paris, working on image denoising using adaptive smoothing methods.

from July to December 2000: intern at Centre National de Recherche Oceanographique de Thiaroye (CRODT), Dakar Sénégal, working on change detection in rainfall data with wavelet methods.

Education

DEA (master) in Mathematics, Computer Vision and Machine Learning of Ecole Normale Supérieure (ENS) de Cachan, Paris, France.

DEA (master) in Applied Mathematics oriented Signal Processing of Université Cheikh Anta Diop (UCAD), Dakar, Senegal.

Maitrise (bachelor) in Applied Mathematics oriented Probabilities and Statistics of Université Gaston Berger (UGB), Saint-louis, Senegal.

Publications

Journal Papers:

- K. Smith, S.O. Ba, D. Gatica-Perez, J.M. Odobez Tracking Attention for Multiple People: Wandering Visual Focus of Attention Estimation, currently under review for IEEE Transactions on Pattern Analysis and Machine Intelligence, Aug 2006

- J.-M. Odobez, D. Gatica-Perez, and S.O. Ba Embedding Motion in Model-Based Stochastic Tracking, IEEE Transaction on Image Processing, vol 15, no 11, Nov 2006

Conference Papers:

- K. C. Smith, S. O. Ba, D. Gatica-Perez, and J.-M. odobez Tracking the Multi-Person Wandering Visual Focus of Attention, International Conference on Multi-modal Interfaces ICMI 06, Banff, Canada, 2-4 Nov 2006

- S.O. Ba and J.M Odobez, A Study on Visual Focus of Attention Recognition from Head Pose in a Meeting Room, 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI06), Washington DC, USA, 1-3 May 2006

- S.O. Ba and J.M Odobez, Head Pose Tracking and Focus of Attention Recognition Algorithms in Meeting Rooms, invited paper at CLEAR'06 Evaluation Campaign and Workshop, Southampton, UK, 6-7 Apr. 2006

- S.O. Ba and J.M Odobez, A Rao-Blackwellized Mixed State Particle Filter for Head Pose Tracking in Meetings, ACM ICMI Workshop on Multimodal Multiparty Meeting Processing (MMMP), Trento Italy Oct. 7, 2005

- S.O. Ba and J.M Odobez, Evaluation of Multiple Cues Head Pose Estimation Algorithms in Natural Environments, International Conference on Multi-media & Expo (ICME), Amsterdam, July 6-8, 2005

- I. MacCowan, M. Hari Krishna,D Gatica-Perez, D. Moore and S.O. Ba, Speech Acquisition in Meetings with an Audio-Visual Sensor Array, International Conference on Multi-media & Expo (ICME), Amsterdam, July 6-8, 2005

- D Gatica-Perez, J.M. Odobez, S.O. Ba, K. Smith and G. Lathoud, Tracking People in Meetings with Particles, invited paper at Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Montreux, Apr. 13-15, 2005

- S. O. Ba and J.M. Odobez, A Probabilistic Framework for Joint Head Tracking and Pose Estimation, International Conference on Pattern Recognition (ICPR), Cambridge, Aug. 23-26, 2004

- C. Kervrann and S. O. Ba, A Non Parametric Approach for Image Restauration with Preservation of Discontinuities, 14 eme Congres Francophone de Reconnaissance de Forme et Intelligence Artificielle (RFIA), Toulouse, Jan. 28-30, 2004

- J.M. Odobez, S.O.Ba and D. Gatica-Perez, An Implicit Motion Likelihood for Tracking with Particles Filters, British Machine Vision Conference (BMVC), Norwich, Sept. 9-11, 2003

Reports:

- S.O.Ba and J.M. Odobez, A Video Database for Head Pose Tracking Evaluation , IDIAP-Com 05-04, Sept. 2005
- S.O.Ba, Filtrage Adaptatif et Restauration d'Images, Rapport de stage du DEA MVA de l'ENS Cachan, Oct. 2002
- S.O.Ba, Ondelettes et Applications en Traitement du Signal, Rapport de DEA du Département de Mathématiques de l'Université de Dakar , 2000

Programming Skills

Operating Systems: Unix, Windows
Programming Languages: C/C++, Matlab

Language Skills

French: fluent speech and writing.
English: fluent speech and writing.

Hobbies

Sport: play football and basketball.
Cultural: like books and movies.