Extraction of an image in order to apply face recognition methods

Nam Jun Pyun

Thèse de doctorat

présentée pour obtenir le grade de docteur
de l'UNIVERSITÉ PARIS-DESCARTES

École Doctorale EDITE

Spécialité : Informatique

Nam Jun PYUN

Extraction d'une image dans une vidéo en vue de la reconnaissance du visage

Soutenue publiquement le 09/11/2015 devant le jury composé de :

Pr. Jean SEQUEIRA	Président
Pr. Liming CHEN	Rapporteurs
Pr. Tien BUI	
Pr. Frédéric PRECIOSO	Examinateurs
Pr. Georges STAMON	
Pr. Nicole VINCENT	Directeur de thèse
Mathieu MARMOUGET	Encadrant, invité
Emmanuèle MOATTI	Encadrante, invitée

Cette thèse s'est déroulée dans le cadre d'une thèse CIFRE entre le laboratoire LIPADE (Laboratoire d'Informatique de PAris DEscartes) dans l'équipe SIP (Systèmes Intelligents de Perception) de l'Université Paris Descartes et l'entreprise Konbini.

Laboratoire LIPADE
Equipe SIP
Université Paris Descartes
45 rue des Saints Pères
75270 Paris cedex FRANCE

Téléphone : (+33)1 83 94 57 41

Konbini
20 rue du Faubourg du Temple
75011 Paris FRANCE

Téléphone : (+33)1 42 46 96 92

Référence BibTEX:

```
@PHDTHESIS{Pyun2015,
  author = {PYUN, N.},
  title = {{E}xtraction d'une image dans une vid\'eo
  en vue de la reconnaissance du visage},
  school = {{U}niversit\'e {P}aris {D}escartes},
  year = {2015}
}
```

Résumé

Une vidéo est une source particulièrement riche en informations. Parmi tous les objets que nous pouvons y trouver, les visages humains sont assurément les plus saillants, ceux qui attirent le plus l'attention des spectateurs. Considérons une séquence vidéo dont chaque trame contient un ou plusieurs visages en mouvement. Ils peuvent appartenir à des personnes connues ou qui apparaissent de manière récurrente dans la vidéo Cette thèse a pour but de créer une méthodologie afin d'extraire une ou plusieurs images de visage en vue d'appliquer, par la suite, un algorithme de reconnaissance du visage. La principale hypothèse de cette thèse réside dans le fait que certains exemplaires d'un visage sont meilleurs que d'autres en vue de sa reconnaissance. Un visage est un objet 3D non rigide projeté sur un plan pour obtenir une image. Ainsi, en fonction de la position relative de l'objectif par rapport au visage, l'apparence de ce dernier change.

Considérant les études sur la reconnaissance de visages, on peut supposer que les exemplaires d'un visage, les mieux reconnus sont ceux de face. Afin d'extraire les exemplaires les plus frontaux possibles, nous devons d'une part estimer la pose de ce visage. D'autre part, il est essentiel de pouvoir suivre le visage tout au long de la séquence. Faute de quoi, extraire des exemplaires représentatifs d'un visage perd tout son sens.

Les travaux de cette thèse présentent trois parties majeures. Dans un premier temps, lorsqu'un visage est détecté dans une séquence, nous cherchons à extraire position et taille des yeux, du nez et de la bouche. Notre approche se base sur la création de cartes d'énergie locale principalement à direction horizontale. Dans un second temps, nous estimons la pose du visage en utilisant notamment les positions relatives des éléments que nous avons extraits. Un visage 3D a trois degrés de liberté : le roulis, le lacet et le tangage. Le roulis est estimé grâce à la maximisation d'une fonction d'énergie horizontale globale au visage. Il correspond à la rotation qui s'effectue parallèlement au plan de l'image. Il est donc possible de le corriger pour qu'il soit nul, contrairement aux autres rotations. Enfin, nous proposons un algorithme de suivi de visage basé sur le suivi des yeux dans une séquence vidéo. Ce suivi repose sur la maximisation de la corrélation

des cartes d'énergie binarisées ainsi que sur le suivi des éléments connexes de cette carte binaire.

L'ensemble de ces trois méthodes permet alors tout d'abord d'évaluer la pose d'un visage qui se trouve dans une trame donnée puis de lier tous les visages d'une même personne dans une séquence vidéo, pour finalement extraire plusieurs exemplaires de ce visage afin de les soumettre à un algorithme de reconnaissance du visage.

Mots Clés : *extraction des yeux, extraction du nez, extraction de la bouche, éléments anatomiques du visage, filtre de Haar, carte d'énergie locale, carte d'énergie globale, analyse multi-seuil, estimation de pose, roulis, lacet, tangage, suivi du visage, suivi des yeux.*

Abstract

The aim of this thesis is to create a methodology in order to extract one or a few representative face images of a video sequence with a view to apply a face recognition algorithm. A video is a media particularly rich. Among all the objects present in the video, human faces are, for sure, the most salient objects. Let us consider a video sequence where each frame contains a face of the same person. The primary assumption of this thesis is that some samples of this face are better than the others in terms of face recognition. A face is a non-rigid 3D object that is projected on a plan to form an image. Hence, the face appearance changes according to the relative positions of the camera and the face.

Many works in the field of face recognition require faces as frontal as possible. To extract the most frontal face samples, on the one hand, we have to estimate the head pose. On the other hand, tracking the face is also essential. Otherwise, extraction representative face samples are senseless.

This thesis contains three main parts. First, once a face has been detected in a sequence, we try to extract the positions and sizes of the eyes, the nose and the mouth. Our approach is based on local energy maps mainly with a horizontal direction. In the second part, we estimate the head pose using the relative positions and sizes of the salient elements detected in the first part. A 3D face has 3 degrees of freedom: the roll, the yaw and the pitch. The roll is estimated by the maximization of a global energy function computed on the whole face. Since this roll corresponds to the rotation which is parallel to the image plan, it is possible to correct it to have a null roll value face, contrary to other rotations. In the last part, we propose a face tracking algorithm based on the tracking of the region containing both eyes. This tracking is based on the maximization of a similarity measure between two consecutive frames.

Therefore, we are able to estimate the pose of the face present in a video frame, then we are also able to link all the faces of the same person in a video sequence. Finally, we can extract several samples of this face in order to apply a face recognition algorithm on them.

Keywords: *eye extraction, nose extraction, mouth extraction, face anatomic elements, Haar filter, local energy map, global energy map, mutli-threshold analysis, pose estimation, roll, yaw, pitch, face tracking, eye tracking,*

Remerciements

Lors de cette thèse, j'ai pu rencontrer de nombreuses personnes, avec qui j'ai partagé de formidables moments. Elle m'a aussi permis de consolider les relations humaines déjà établies.

Avant tout, je tiens à remercier l'entreprise Konbini qui m'a accompagné pendant toute la durée de la thèse. Tous ses membres m'ont accueilli chaleureusement, facilitant mon intégration au sein de leur équipe de recherche et développement, mais aussi en créant un environnement propice à l'évolution de mon travail. En particulier, je remercie mes encadrants au sein de l'entreprise Mathieu Marmourget et Emmanuèle Moatti, ainsi que tous les membres de l'équipe scientifique Vincent, Cendrine et Thomas.

D'autre part, je tiens à remercier tous les membres du LIPADE, en particulier, ceux de l'équipe SIP pour toute leur attention et leurs conseils si précieux. Tout d'abord, je souhaite adresser mes remerciements aux professeurs de l'équipe, Georges Stamon, Laurent Wendling, Florence Cloppet, Camille Kurtz, Nicolas Loménie et plus particulièrement ma directrice de thèse Nicole Vincent. Je souhaite remercier aussi tous mes amis et autres collègues du laboratoire, Rabi, Nicolas, Hassan, Arnaud, Hee-Chang, Marwen, Adrien, Mickaël, Maya, Sameh, Adam, Soumaya, Michaël, Héloïse et Cuong. Grâce à eux, le laboratoire a toujours été un lieu convivial où l'entraide et les sourires en étaient les piliers.

Par ailleurs, je tiens à remercier toute ma famille, en premier lieu, mon épouse Heeyun, mon père, ma mère, mon frère, ma belle-sœur, ma petite nièce, ma cousine Sujin. Je n'oublie pas non plus mes beaux-parents, mes oncles et tantes, mes cousins. Leur soutien, leur affection ont toujours été et resteront la source de toute motivation. Je pense aussi très fort à toi, Nanou. Tu nous manque tellement et j'espère que tu es fière de moi.

Enfin, je souhaite remercier plus particulièrement ma directrice de thèse, le professeur Nicole Vincent. Merci pour votre disponibilité. Merci pour ce soutien permanent que ce soit dans le domaine scientifique ou dans ma vie personnelle. J'ai particulièrement apprécié toutes les heures de discussion que nous avons pu partager. Sans nul doute, vous avez été le guide que tout doctorant recherche.

Contents

Résumé **iii**

Abstract **v**

Remerciements **vii**

Contents **ix**

List of Figures **xi**

List of Tables **xvii**

1 Introduction **1**
1.1 Thesis context . . . 3
1.2 Motivation . . . 3
1.3 Overview of this thesis . . . 6

2 Human face salient element extraction **7**
2.1 Introduction . . . 9
2.2 State of the art . . . 11
2.2.1 Wavelet based Features . . . 11
2.2.2 Linear features . . . 16
2.2.3 Features based on Statistics . . . 16
2.2.4 Shape features . . . 17
2.2.5 Template based features . . . 18
2.2.6 Knowledge based features . . . 18
2.3 Motivation . . . 20
2.4 The face salient element extraction method . . . 25
2.4.1 Energy from Haar-like features . . . 25
2.4.2 Overview of face anatomic elements extraction . . . 34
2.4.3 Extraction of face vertical contours . . . 38
2.4.4 Eyes, nose tip and mouth extraction . . . 49
2.4.5 Multi-threshold analysis of the normalized horizontal energy map . . . 65
2.5 Evaluation . . . 71
2.5.1 Still face image databases . . . 71
2.5.2 Separating face area from the background . . . 71
2.5.3 Anatomic region extraction evaluation . . . 74
2.6 Conclusion of face salient element extraction . . . 80

3 Head pose estimation using Haar energy and face salient elements **81**
3.1 Introduction . . . 84
3.2 Head pose estimation state of the art . . . 86
3.2.1 Methods based on templates . . . 86
3.2.2 Methods using classification . . . 87
3.2.3 Geometric methods . . . 88
3.2.4 Methods using flexible models . . . 89
3.2.5 Nonlinear regression methods . . . 90
3.2.6 Methods based on Embedding . . . 91
3.2.7 Hybrid methods . . . 92
3.2.8 Discussion . . . 92
3.3 The proposed face pose estimation methods . . . 94
3.3.1 Estimation of the roll . . . 94
3.3.2 Yaw and pitch estimation . . . 103
3.4 Evaluation of our pose estimation method . . . 112
3.4.1 Evaluation of the roll estimation method . . . 112
3.4.2 Evaluation of yaw and pitch estimation method . . . 117
3.4.3 Test database . . . 117
3.4.4 Parameters . . . 117
3.5 Conclusion of pose estimation . . . 121

4 Face Tracking **123**
4.1 Introduction . . . 125
4.2 Face tracking state of the art . . . 126
4.2.1 Motion based methods . . . 126
4.2.2 Model based approaches . . . 128
4.3 Face Tracking Method . . . 131
4.3.1 Finding the eyes at the first face sample . . . 131
4.3.2 Tracking the region containing both eyes . . . 132
4.3.3 Similarity functions . . . 133
4.3.4 The search area . . . 137
4.3.5 Selection of the best samples for the purpose of face recognition . . . 138
4.4 Evaluation . . . 144
4.4.1 YouTube Faces database . . . 144
4.4.2 Choosing the similarity function . . . 145
4.4.3 Tracking results . . . 145
4.5 Conclusion . . . 149

5 General conclusion and perspectives **151**
5.1 General conclusion . . . 152
5.1.1 Back to our face salient region extraction method . . . 152
5.1.2 Back to our head pose estimation methods . . . 153
5.1.3 Back to our tracking method . . . 153
5.1.4 Finding the best face samples . . . 154
5.2 Perspectives . . . 155
5.2.1 Applications of our work . . . 155
5.2.2 Continuation of our work . . . 155
5.2.3 Extension to other objects . . . 156

Personal Publications **157**

Bibliography **159**

List of Figures

1.1 Some images where there are shoes. 5
1.2 Overview of this thesis giving the best face samples among those present in a video sequence. 6

2.1 Difficulties of extracting face. 9
2.2 Haar-like features used by Viola and Jones. 12
2.3 Scheme of image I with a rectangular area ABCD contained in I. 13
2.4 Cascade of classifiers. 15
2.5 Face on a frame of a video sequence as well as its related control points. 21
2.6 Coplanar triangulation: left and right images are two different projections of the same 3D result. 23
2.7 Correct triangulation: left and right images are two different views of the same result. 23
2.8 Studied Haar Patterns. 25
2.9 The face example used in this section. 26
2.10 Energies of horizontal Haar filter according to the filter width w and height h in pixels. The face window width and height equal both to 364 pixels 27
2.11 Binarized horizontal energy maps. Thresholds are chosen manually with a visually suitable value to illustrate. 28
2.12 Normalized energy maps of vertical Haar filter according to the filter width w and height h in pixels. The face window width and height equal both to 364 pixels. 30
2.13 Binarized vertical energy maps. Thresholds are chosen manually with a visually suitable value to illustrate. 31
2.14 Binarized diagonal energy maps. Thresholds are chosen manually with a visually suitable value to illustrate. 31
2.15 Linear combination of horizontal and vertical Haar energies. the result is an edge detector. 33

2.16 Visual results of non-linear combination of horizontal and vertical energies. Values which belong to S_{hori} is represented in green, values of S_verti is represented in red and other values of the combination (belonging to S_{other}) in blue. . . 35
2.17 Global view of face salient anatomic element extraction. . . 36
2.18 Scheme of vertical face borders extraction, the result is a mask which separates the face from the background. 37
2.19 Extraction of face salient elements extraction. 38
2.20 The first line contains the original images, whereas the second one contains normalized vertical energy maps. 39
2.21 The first line contains the original images, The others contain binarized vertical energy maps according to the threshold t. 41
2.22 Cumulative histograms in percentage of vertical energy map. Original images are those of Figure 2.21. 42
2.23 Binarized vertical energy map using adaptive threshold. . . 43
2.24 Connected component bounding boxes the area of which is more than 1% of the area of the greatest bounding box. . . 44
2.25 Number of pixels belonging to connected component bounding boxes according to the image column. 45
2.26 Selected left and right vertical borders; the left vertical border is represented in green boxes and right one is represented in red boxes. 47
2.27 Schematic example of a selected left vertical border; the left side of this figure shows the connected components of this border, the right side shows the mask. 48
2.28 Generated graph and selected connected components from the example of Figure 2.27. 49
2.29 Vertical border face masks. 50
2.30 Haar horizontal mirror pattern of width w and height h. . 50
2.31 From top to bottom: original face window, detected face mask, normalized horizontal energy map of the original image, normalized horizontal energy map after applying the face mask. 52
2.32 Examples of energy map En_{Hh} binarization according to threshold t without the face mask and after applying the face mask. 53
2.33 Bounding boxes of connected components according to threshold t. 55
2.34 **Left:** Scheme of some basic knowledge used in our method. **Right:** An anatomic region bounding box can be represent by the upper left point S and the lower right point T. . . 56

2.35 Histogram of widths of connected component bounding boxes on y-axis, respectively for face window 'a', 'b', 'c' and 'd'. . 57
2.36 Candidate anatomic regions obtained after merging close candidate regions. four candidate anatomic regions are extracted from this histogram. 59
2.37 Histogram of occurrences of connected component bounding boxes on x-axis. 60
2.38 a) Initial connected components and their bounding boxes of candidate anatomic region containing both eyes. b) Number of connected component pixels according to the abscissa as well as the upper level set. c) Separation of left and right candidate anatomic regions using this level set. 61
2.39 Connected components taken into account for a fixed threshold: they should belong either to eyes or to nose tip or to mouth. 63
2.40 Examples of noticeable connected components: the highest one, CC_{nose} is indicated by a blue arrow and the widest one, CC_{mouth} is indicated by a red arrow. 64
2.41 Extracted candidate anatomic regions of both eyes, nose and mouth for a fixed binarisation threshold of 0.75 of the normalized horizontal energy map. 65
2.42 Candidate anatomic regions extraction according to threshold t applied on normalized horizontal energy map. 66
2.43 Variation of position and size of the bounding boxes of left eye candidate anatomic region for the face of image 'd'. . . 68
2.44 Functions D and $w_R \times h_R$ associated with the left eye region area of image 'd' according to the threshold t. 70
2.45 Selected candidate anatomic regions: they are the candidate anatomic regions for which we have computed suitable thresholds. 70
2.46 Examples of good masks. The first line contains images from Color Feret database and the second line contains images from BioID database. 72
2.47 Examples of incomplete masks. The first line contains images from Color Feret database and the second line contains images from BioID database. 73
2.48 Examples of wrong masks. All images belong to Color Feret database. 73
2.49 Percentage of chosen thresholds by the multi-threshold analysis. 76
2.50 Visual results on BioID. Incomplete or wrong extractions are in last line. 78

2.51 Visual results on Color FERET. Same disposition as Figure 2.50. 79

3.1 Visualization of the roll, yaw and pitch of a head. 84
3.2 Face salient extraction results according to roll angle α. . . 95
3.3 Image a illustrates a horizontal straight line with the horizontal Haar filter. Image b illustrates an oblique straight line with the horizontal Haar filter centered in A. In (a), the energy is the area of the green zone. In (b), the energy is the difference of the areas of the blue and green zones. . . 96
3.4 Original face window and application of the circular mask on the face window. 97
3.5 Graph of the global horizontal energy of Figure 3.4a according to the rotation α. In the original image, the roll of the face is almost null. 98
3.6 Graph of the global horizontal energy of Figure 3.4b according to the rotation α. In the original image, the roll of the face is almost null. 99
3.7 Global scheme of the roll estimation method. 100
3.8 Some bounding boxes of extracted salient face elements according to the threshold applied on local horizontal energy map and the rotation angle α. 102
3.9 Schematic view of eye image plan projection. 104
3.10 Examples of faces depending on the yaw interval. 106
3.11 Schematic representation of the extracted bounding boxes in left profile and diagonal views, in null yaw view and in right diagonal and profile views. 107
3.12 Parameters extracted from face salient element bounding boxes. 108
3.13 Distribution of face images (%) according to the difference operator Δ. 114
3.14 Correctness (%) according to Δ. 114
3.15 Correctness (%) according to the rejection threshold T. . . 115
3.16 Distribution of face images (%) according to the absolute error in Color Feret database. 116
3.17 Logarithm of the distribution of face images according to the absolute error in Color Feret database. 117

4.1 Scheme of the tracking process based on similarity of energy maps. 132
4.2 Scheme of the tracking process based on similarity of binarized energy maps. 134
4.3 Approximate face proportion. 140

4.4 Frame examples in YouTube database. 144
4.5 Tracking error with very small eye region 146
4.6 Tracking error with total occlusion. 146
4.7 Tracking of other objects. 147
4.8 Selection of the five best samples in the video clip. 148

5.1 Overview of this thesis giving the best face samples among those present in a video sequence. 151

List of Tables

2.1 Examples of normalized eigenvalues according to 3D reconstruction point distribution. 24
2.2 Adaptive threshold obtained on images of Figure 2.21. . . 43
2.3 Selected thresholds respectively for the left eye, right eye, nose tip and mouth regions of images 'a', 'b', 'c' and 'd'. . 70
2.4 Evaluation of the mask generation method. The mask should separate the face region from the background. 74
2.5 Comparison of Li et al. and Asteriadis et al. methods with ours on BioID database. 76
2.6 Correctness and relative mean-error of candidate anatomic regions with a fixed energy map binarization threshold as well as the multi-threshold approach. 77
2.7 Nose and mouth detection rate. 78

3.1 Yaw estimation according to E and to N. "ND" means "not determined". 109
3.2 Distribution of faces and correctness according to the number of extracted local maximums in Color Feret. 113
3.3 Correctness(%), mean absolute error and standard deviation of the roll estimation in Color Feret and BioID. 116
3.4 Constants value used for yaw estimation, as well as the experiments estimation of the yaw. 118
3.5 Element detection rate according to left yaw variation and rate of well classified left yaw per yaw class. 119
3.6 Confusion matrix of yaw estimation 119
3.7 Elements detection rate according to pitch 120
3.8 Confustion matric of pitch estimation 120

Chapter 1

Introduction

Chapter contents

1.1 Thesis context 3
1.2 Motivation 3
1.3 Overview of this thesis 6

Chapter summary

Because of the expansion of the Internet and of the storage capabilities, digital videos are used worldwide. At this point, digital videos represent more than 60 % of the data traffic on the web (streaming, uploads and download). Moreover, a single video has an important amount of images, Hence, it is impossible to index and annotate manually all the images included in videos.

Computer vision can help to achieve these tasks with automatic or semi-automatic systems. In this chapter, we will present the thesis context, the aim and motivation of our work.

Résumé du chapitre

La vidéo sous forme numérique est omniprésente grâce à l'expansion du web et des capacités de stockage. A l'heure actuelle, les vidéos représentent plus de 60% du trafic sur le web (streaming, téléchargements). De plus, une simple vidéo contient un nombre très important d'images. Ainsi, il est impossible d'indexer et d'annoter manuellement toutes les images incluses dans les vidéos.

La vision par ordinateur permet toutefois l'annotation semi-automatique ou automatique de ces images. Dans ce chapitre, nous présenterons le contexte de la thèse ainsi que le but et les motivations de notre travail.

1.1 Thesis context

This thesis is the result of a partnership between the SIP (Systèmes Intelligents de Perception) team of LIPADE (Laboratoire d'Informatique de PAris DEscartes) and the company Konbini. It is a CIFRE (Conventions Industrielles de Formation par la REcherche) thesis. In this section, we will first present the context of this thesis, followed by the motivations. This section will end by an overview of this dissertation.

One of the activities of Konbini is the organization of events for other companies or brands. For example, parties can be organized for the launching of a new product.

Another activity is the promotion of a client's product. Konbini can create commercials for televisions, but it also proposes to broadcast these commercials in other platforms such as in social networks or in Youtube.

Konbini also handles a website. The target of this website is the adolescents or young adults, from 15 to 35 years old. The company broadcasts various cultural news. The content is various: it can be about music, cinema, fashion or headlines. There are many video interviews, music clips...

Konbini is also a video production company. It creates videos, such as interviews, for clients in many platforms, media sources or television channels.

Finally, Konbini also handled a web television where they broadcasted many videos created either by Konbini or by other partners.

1.2 Motivation

Konbini has a large amount of digital videos. These videos are produced either by Konbini or by other partner companies. One of the aims of Konbini research is annotating the videos with the objects present in their frames. Konbini wants to create a new player able to give the positions of all included objects. Indeed, traditionally, the only information we have from a digital video is in its meta-data. Among these meta-data, we have the description of the video's content. However, if these global meta-data can be enough to describe a still image, they cannot describe accurately a video because of the important amount of images and thus of information. Actually, the meta-data of a video is only a short summary of the content and not a real description.

Konbini wants to create a player which includes some tags, each one

associated with the represented object and its position in the video frame. For example, Konbini had to annotate a specific car in a video. All the car positions and sizes were labelled manually frame by frame. Konbini wants a video player able to annotate any object. This is a challenging aim. However, computer vision techniques are not yet mature enough for finding all kinds of objects. Actually, even a simple object is difficult to find in all images where it is present because of scale, illumination, pose and other variations.

As shown in image 1.1, a simple object like shoes can be difficult to detect. In image b and c, shoes are visually small whereas in image a, d or e, shoes are represented in a higher scale. As a consequence, their appearance changes a lot. Some extreme lightning conditions, like in image e, make the detection of shoes difficult. In image b, shoes are blurred because of the movement of the feet. Some of the shoes are frontal views and other are profile views, their appearance changes according to the camera and object positions. Moreover, all these images contain shoes, but if we can reasonably assume that shoes are an important object in image a, d and e, they seem less important in image b and almost negligible in image c.

Hence, it is very difficult to create a hierarchy of objects, because the importance of an object depends on the video context and on human subjectivity. Moreover, annotating all objects of a video frame is counterproductive; we will surely have too much information.

So, at this point, the question to be asked is: which object should we annotate? Indeed, annotating all the objects, frame by frame, is time consuming for a poor relevance, because many of them can be negligible in the context of the video. Moreover, If too many objects are labeled without a hierarchy on them, salient information of video frames will be gloss over because of the amount of detected objects. Hence, we must restrict the objects we want to find on video frames.

Among all the objects of the world, the human face is a singular object. Indeed, everybody will consider human faces as salient objects. As human beings, our eyes focus naturally to human faces present in the images. Therefore, we choose to focus on analyzing human faces in video.

Detecting faces in video frames is the first task. Although some approaches achieve this detection with frontal or almost frontal faces, most of them cannot detect faces with a different pose. Here, our goal is not to recognize a face, but it is to extract best samples of face images on which recognition should succeed.

To find these samples, we need some facial features. These features must be robust to illumination, scale, but also to pose variations. We

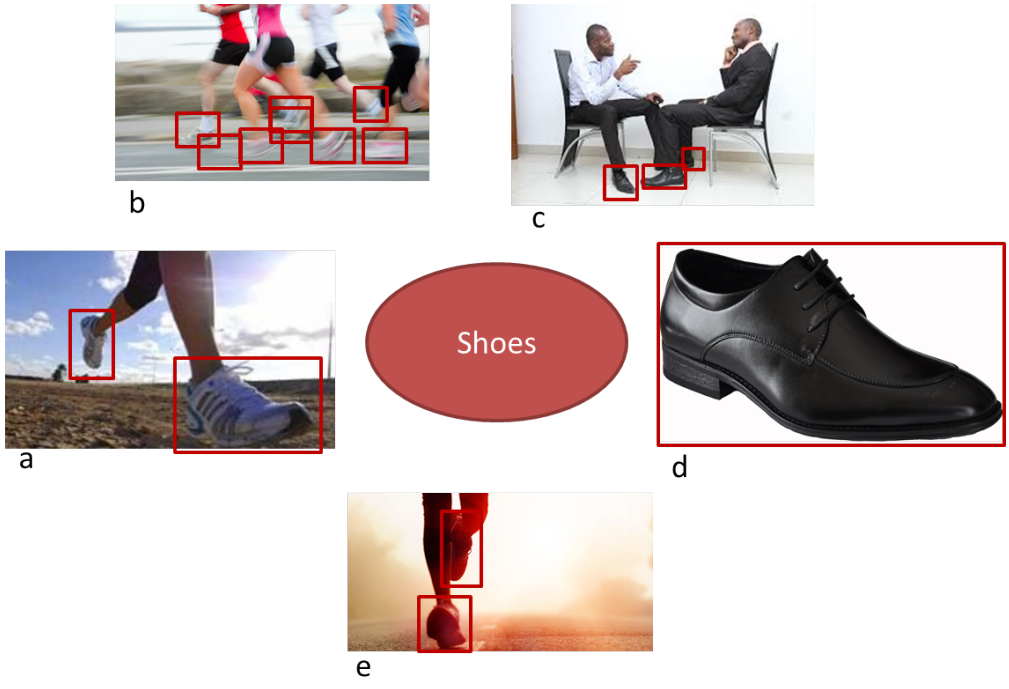

Figure 1.1: Some images where there are shoes.

assume that in the whole sequence where a face is present, at least one sample of this face is detected by face detector. From this sample, we have two other tasks to do to give the best samples for recognition.

- Once facial features have been extracted, we should be able to estimate the face pose. We assume that best samples are those with null roll, yaw and pitch values. So, having the pose of each face sample from a video sequence will respond to the question: which are the best samples?

- Finally, we also have to track the face. Indeed, since only one face sample should be detected, we have to propagate this face detection to the other frames of the video sequence. Hence, we will no longer need to detect face samples in other frames. Moreover, some face samples which are not detected by the face detector because of the pose should still be extracted by the tracking.

1.3 Overview of this thesis

We propose new facial features extracted on energy maps based on Haar-like filters. Then, using these features, we extract salient face regions: left and right eyes, the nose basis and the mouth. This method will be presented in chapter 2

Once salient regions have been extracted from face, in order to extract the most frontal sample of this face in a video sequence, we present in chapter 3 a head pose estimation method which can be actually divided into two methods. First, the roll is estimated using the energy maps. Then, the yaw and the pitch are estimated using the relative positions and size of the extracted salient regions.

Finally, in chapter 4, we present a tracking method. We assume that eyes are the most representative regions of the face. Hence eyes are tracked. From the positions and sizes of eyes, we can approximately estimate the position and size of the whole face and hence estimate the head pose using methods presented in the previous chapters.

Figure 1.2 shows an overview of the whole system presented in this dissertation.

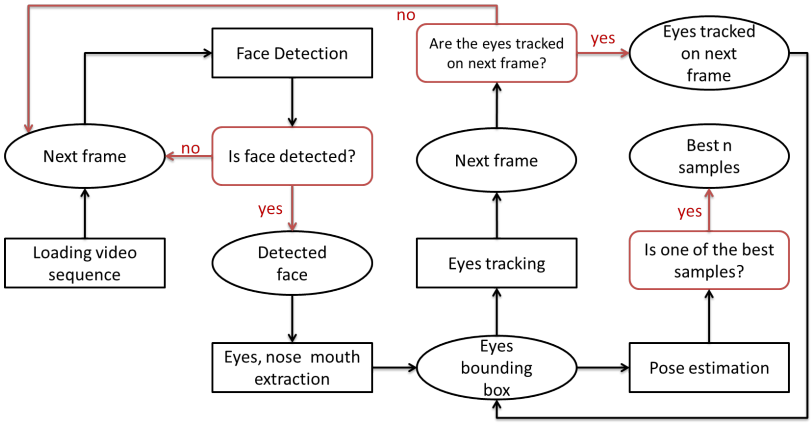

Figure 1.2: Overview of this thesis giving the best face samples among those present in a video sequence.

Chapter 2

Human face salient element extraction

Chapter contents

2.1 Introduction 9

2.2 State of the art 11

2.2.1 Wavelet based Features 11

2.2.2 Linear features 16

2.2.3 Features based on Statistics 16

2.2.4 Shape features 17

2.2.5 Template based features 18

2.2.6 Knowledge based features 18

2.3 Motivation 20

2.4 The face salient element extraction method . . 25

2.4.1 Energy from Haar-like features 25

2.4.2 Overview of face anatomic elements extraction . 34

2.4.3 Extraction of face vertical contours 38

2.4.4 Eyes, nose tip and mouth extraction 49

2.4.5 Multi-threshold analysis of the normalized horizontal energy map 65

2.5 Evaluation 71

2.5.1 Still face image databases 71

2.5.2 Separating face area from the background 71

2.5.3 Anatomic region extraction evaluation 74

2.6 Conclusion of face salient element extraction . 80

Chapter summary

Extracting eyes, nose and mouth in a face is still a challenge in pattern recognition. Finding the position of these anatomic elements may be the first step to achieve many tasks, such as segmentation, recognition or identification, head pose estimation, landmarks localization, facial expression detection and face tracking...

A method based on analysis of horizontal direction elements, on adaptive horizontal Haar-like features including spatial relation knowledge is proposed. Here, we assume faces have been detected in a sub-window giving the scale. In order to locate these salient areas in faces, a horizontal energy map is computed. To overcome the illumination variations, this method includes a multi-threshold analysis. This detector has been tested on Color Feret, BioID face databases.

Résumé du chapitre

Extraire les yeux, le nez et la bouche d'un visage est toujours, à l'heure actuelle, un défi dans le domaine de la reconnaissance des formes. Trouver la position de ces éléments anatomiques peut être la première étape permettant de réaliser de nombreuses tâches, comme la segmentation, la reconnaissance ou l'identification, l'estimation de pose du visage, la localisation de points d'intérêt, la détection de l'expression faciale, le suivi du visage... Nous proposons, ici, une méthode basée sur l'analyse des éléments horizontaux, sur des filtres de Haar horizontaux adaptatifs et sur certaines connaissances des relations spatiales entre les éléments du visage. Nous supposons que les visages sont détectés dans une fenêtre dont la taille nous donne leur échelle. Afin de localiser ces éléments saillants du visage, une carte d'énergie horizontale est calculée. Pour surmonter les variations d'illumination, cette méthode inclut une analyse multi-seuils. Nous avons testé la méthode d'extraction des éléments saillants sur les images des bases Color Feret et BioID.

2.1 Introduction

To manipulate face images, we need to extract features from these images. The nature of these features varies a lot. First, we can wonder whether the features are local or global. In the field of face feature extraction, methods are almost all local. In particular, in face detectors, methods are bottom up; local features are gathered to make a higher level and more global decision. Some methods use shape information. Other use textures, colors, salient points or a combination. Many applications use face detection in social networks or face detection systems are directly incorporated in cameras. The detectors achieve their tasks in still images with a good recall and accuracy. However, in such images, faces are often taken in good external conditions, making the detection easier to achieve. In videos, conditions vary a lot; the task is not so simple. Moreover, objects in videos often move (because of camera movement or object movement itself). Faces are no longer stable and are often blurred.

Figure 2.1: Difficulties of extracting face.

As shown in Figure 2.1, many difficulties may occur in detecting faces.

- Variations of pose: face images are projections of a 3D object. According to pose, the appearance varies.
- Variations of scale: size of the face images varies. Especially, when scales are low, parts of the face are no longer separated.

- Variations of illumination: face illumination may vary from an image to another.
- Variations of patterns: face is a deformable object. A subject can smile or not. Moreover, eyes can be closed.
- Occlusions: Hands, glasses, hats, mustaches, beards can occlude partially the face.

The proposed method should aim to extract higher level information as face features: eyes, nose tip and mouth. The next section presents the state of art in face feature extraction. Then, the proposed method is described. Finally, it is evaluated on several face image databases.

2.2 State of the art

Extracting salient features in faces is relevant for many applications, such as face recognition [Jain et Unsang, 2009], head pose estimation [Murphy-Chutorian et Trivedi, 2009], face tracking [Zhou et al., 2010a] or facial expression [Zhong et al., 2012]. In particular, locating eyes, nose and mouth may be useful information. For example, in order to track a face in a video, many methods use Active Appearance Models (AAM) [Cootes et al., 2001], since they are quite efficient and accurate. The first step in AAM involves learning the global deformation of face by applying a PCA on salient points localization and intensity of a face (e.g. eye, nose, mouth corners and boundary points). The second consists in fitting a given set of salient points from a face with the model. The main drawback of AAM is salient points need most often to be placed manually. To achieve making AAM fully automatic, finding location of these landmarks is required. Although giving the location and the bounding boxes of eyes, nose and mouth is not enough to find salient points, it may improve accuracy, since it delimits the search area. There are many features extracted from human faces in face detection issue. They can be categorized in different sets of features:

- wavelet based features,
- linear features,
- statistical features,
- shape features,
- template based features,
- knowledge based features.

2.2.1 Wavelet based Features

The first set is composed by approaches based on wavelets. Here, the main idea is to extract local features in an appropriate subspace using machine learning techniques, such as in [Vukainovic et Pantic, 2005] where Gentle-Boost is used on Gabor features or in [Akhloufi et Bendada, 2010] where PCA is used on texture features called Local Ternary Patterns (LTP). LTP is presented as a generalization of LBP[Tan et Triggs, 2010]. In order to understand how this kind of methods works, it is useful to describe Viola and Jones face extractor.

2.2.1.1 Back to Viola and Jones face detector

Viola and Jones face detector is based on three main steps. First, it extracts local descriptor from Haar-like features by using integral image to speed up the process, and then it uses a learning algorithm which is, actually, a feature selection step. Finally, an additional cascade of classifiers is applied which enables the detector to work in real-time. Haar-like features are actually filters, as shown in figure 2.2.

Figure 2.2: Haar-like features used by Viola and Jones.

They can be seen as convolution filters where white area contains value 1 and black area contains value -1. By varying its width and its height, a large number of filters are generated from this short list of initial patterns. According to the observation scale and the pattern, Haar-like features can describe texture (e.g. pattern with small size), as well as higher level information such as shape (e.g. large pattern). So, Haar-like features are a descriptor which describes the general appearance. Hence, Haar-like features gather many kind of information. Moreover, although they describe characteristics of different nature, they are expressed by the same formulation. Therefore, it is possible to compare or combine them. However, there is an important drawback; the number of generated features is very large. Some of them are redundant whereas others are irrelevant.

Given the intensity of the pixel $I(x, y)$ at position (x, y) of an image I and given a Haar filter H, the value $f_H(x, y)$ is computed by the formula 2.1.

$$f_H(x, y) = (I * H)(x, y); \tag{2.1}$$

For a given pattern H, computing one Haar single value requires $w \times h$ accesses in the image I intensity table, where w and h are the width and height of Haar pattern H. So, Computing Haar feature of the whole Image I depends on the width and height of I, as well as the width and height of the pattern H. In such case, a real-time computation is not possible. To make the face extractor fast, Viola and Jones introduce the integral image which is simply a summed area table. The integral image II of image I is given by formula 2.2.

$$
\begin{aligned}
II(X,Y) &= \sum_{0\leq x\leq X}\sum_{0\leq y\leq Y} I(x,y) \\
II(X,Y) &= II(X-1,Y)+II(X,Y-1) \\
&\quad -II(X-1,Y-1)+I(X,Y)
\end{aligned}
\tag{2.2}
$$

As we can see, the computation of each value in the integral image needs only three references in II and one in I. In other words, each value of the integral image is computed in constant time. So the computation of the whole integral image depends only on the width and height of the image I. Once the integral image has been obtained, the computation of the sum of intensities of any rectangular area is made in constant time too with respect to Haar pattern size.

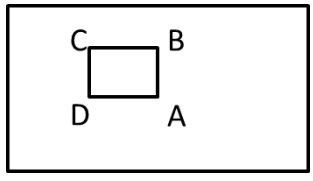

Figure 2.3: Scheme of image I with a rectangular area ABCD contained in I.

Figure 2.3 shows the image of a rectangular area $ABCD$ included in the image I. Once the integral image II has been computed, the computation of the sum of all intensities included in the rectangle $ABCD$ is given by the formula 2.3.

$$
\begin{aligned}
\sum_{(x,y)inABCD} I(x,y) &= II(A)+II(C) \\
&\quad -II(B)-II(D)
\end{aligned}
\tag{2.3}
$$

Formula 2.3 shows that only four references to the integral image are needed to compute the sum of all intensities of any rectangular area of I. As all Haar-like patterns are combinations of rectangular areas, each value of Haar feature is computed in constant time and no longer depends on the pattern width, nor its height. For instance, the first pattern of Figure 2.2 consists of two rectangles. As we show, four references of II are required for each rectangles and thus eight references to II are needed, but since two vertices of both rectangles are common, only six references of the integral image II are actually required.

The second step in Viola Jones face detector is a learning step using a modified version of AdaBoost algorithm. In Viola and Jones method, the boosting algorithm both selects the best features and train the classifier. AdaBoost is used to generate a strong classifier as a weighted combination of weak classifiers. Each weak classifier is associated with a single Haar-like feature. The classifier is considered as "weak", because even the best feature is not expected to classify the whole correctly. In practice, the best feature may only classify correctly 51% of a given database. However, it is assumed that a weighted combination of weak classifiers gives better results than separated ones. In Viola and Jones method, in order to train the classifier, 24×24 pixels sub-windows are used.

The learning set consists of sub-windows containing faces and non-face image part. Even if there are only a few patterns, as the size may vary, more than 160 000 different patterns can be associated with a single 24×24 sub-window. It is easy to understand why selecting features among 160 000 features is essential. As shown in formula 2.4, each weak classifier $h(x, f, p, \theta)$ is associated with the Haar function f, the position x, the sign of Haar function p and a threshold θ.

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & otherwise \end{cases} \tag{2.4}$$

First, each weak classifier is associated with the same weight. An iterative algorithm selects at each iteration t the best weak classifier h_t by minimizing its error rate in the whole training set. Then each weight α_t is adjusted to give more importance to weak classifiers with low error rate. Finally, a strong classifier $C(x)$ is built as a combination of weighted selected weak classifiers (formula 2.5)

$$C(x) = \begin{cases} 1 & \text{if } \sum_T \alpha_t h_t(x) \geq \frac{1}{2} \sum_T \alpha_t \\ 0 & otherwise \end{cases} \tag{2.5}$$

T is the number of iterations and the number of selected weak classifiers. Only 200 feature weak classifiers are needed to achieve good detection.

The last contribution of Viola and Jones face detector is the cascade of classifiers. The cascade of classifiers should aim at achieving good performance while reducing the computation time. In order to test the presence of faces in a given image, all sub-windows of different sizes generated in this image must be tested. Viola and Jones method does not need to build a pyramid of images of different scale, because of the integral image. However, compared to the great number of sub-windows originally generated from one single image, sub-windows which contain a face in a suited scale

are extremely rare. In other words, the large majority of sub-windows do not contain face in a suited scale. The cascade of classifiers uses this information. For example, a strong classifier generated with the two weak classifiers with the lowest error rates has a recall of 100% but a false positive rate of 50%. Of course, this classifier can not be used as a detector. However, since the sub-windows of face are extremely rare, such classifier with only two features can eliminate 50% of sub-windows which do not represent a face. Obviously, this classifier is both simple to train and fast when testing a sub-window. The cascade of classifiers is composed of a sequence of classifiers (Figure 2.4) where the classifier in position $n+1$ has a lower false positive rate but a higher computation time than the one in position n.

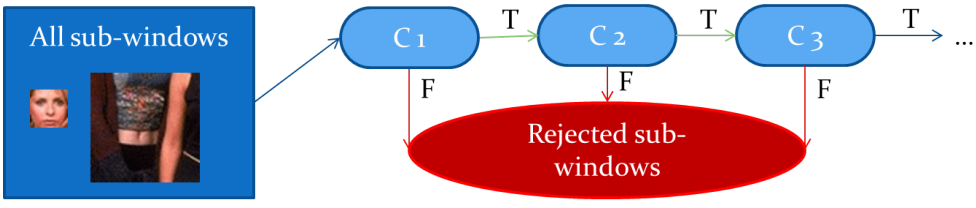

Figure 2.4: Cascade of classifiers.

Even if a face sub-window must be validated as a face by all the classifiers of the sequence, such technique is efficient, because it enables to reject most of the sub-windows which are not faces in the fast first levels of the cascade.

Viola and Jones face detector is actually a good example of how an appearance based face detector can be made. This kind of techniques needs to generate a great number of local features, to train them, in order to discriminate face sub-windows from the others. In other words for appearance based methods, we need to ask ourselves two questions. The first one is which descriptor is used and the second one is which learning technique is used.

2.2.1.2 Extensions of Viola and Jones face detector

Extensions of Viola and Jones method have been developed. In [Lienhart et Maydt, 2002], the authors extend the initial set of features by adding 45 degree rotated patterns. To compute Haar functions, they introduced a new rotated integral image which is computed in only two passes of the whole image. In [Li et al., 2002], since the initial Haar-like feature set is not suited for multi-view face detection, the authors introduced more

free features. The rectangular areas in a given Haar-like feature have a flexible size $x \times y$ and are separated by a distance of (dx, dy), in order to take into account non-symmetrical view which appears when faces are no longer frontal. In [Jones et Viola, 2003], the authors also proposed their own extensions which integrate multi-view issue by adding new patterns composed of overlapped and shifted rectangular areas. In [Brubaker et al., 2008], a Classification And Regression Tree (CART) is used. They show that CART based weak classifiers achieve face detection with better results. In [Wu et al., 2004], the authors proposed to use the histogram of feature values associated with each Haar-like feature in a RealBoost algorithm.

2.2.2 Linear features

In [Meynet et al., 2007], the authors use anisotropic Gaussian filters which are actually combination of a Gaussian in one direction and its derivative in the orthogonal direction. Then, several transformations are applied of these filters such as translation, rotation, bending or anisotropic scaling. These transformations will generate a large number of functions. Similarly to Viola and Jones face detector, AdaBoost and a cascade of classifiers is used to train and to make the detection faster. In [Xiangrong Chen et al., 2001], an extension of non-negative matrix factorization (NMF), called local NMF (LNMF) is used to generate features from face. AdaBoost is then used to select the most significant features among all generated by LNMF. In [Wang et Ji, 2005], the authors combine Fisher discriminant analysis and AdaBoost to improve the accuracy of weak classifiers. They propose a recursive scheme for non-parametric discriminant analysis (RNDA). First, with such schemes, they assume training step should be shorter, since only a subset of the whole feature set is used. Moreover, they assume that RNDA can improve the face detection despite pose variation. These features improve the speed of the convergence of classifiers. However, they are usually longer to compute than Haar features or LBP.

2.2.3 Features based on Statistics

Histograms of local features are also widely used in face detection. In [Ojala et al., 2002], the authors introduced a new feature called local binary patterns. A LBP value is obtained by comparing the value of a central pixel g_c to the value of the pixels in the neighborhood $\mathcal{N}_{R,P}$ of g_c, according to two parameters, the number of neighbors P and the distance R between the central pixel and its neighbors g_i, $i \in [0, P-1]$ (formula 2.6 and 2.7).

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.6}$$

$$LBP_{P,R} = \sum_{p \in \mathcal{N}_{R,P}} s(g_p - g_c) 2^p \tag{2.7}$$

$LBP_{P,R}$ is an integer which varies between 0 and 2^{P-1}. In [Jin et al., 2004], LBP are used as features to detect faces. A multivariable Gaussian Model is used and face and non-face images are then classified under a Bayesian framework, whereas in [Zhang et al., 2007a], LBP are still used to detect faces with a boosting training step. LBP show also good results in face recognition task such as in [Ahonen et al., 2004]. In [Yan et al., 2008a], the authors present the locally assembled binary feature which use both modified Haar-like features and LBP. Their method shows good results on CMU/MIT frontal face database.

In [Levi et Weiss, 2004], local edge orientation histograms are proposed. First, a Sobel mask is applied on the face window to extract the gradients. The magnitude and the orientation of the gradients are computed. Both magnitude and orientation are used in a histogram, the features depending on the orientation. These features are then trained in an AdaBoost algorithm. In [Dalal et Triggs, 2005], a similar histogram (histograms of oriented gradients or HoG) based approach is proposed. In [Wang et al., 2009], HoG and LBP are combined to detect human.

These methods suffer from the necessity of a learning step including a database with a ground truth. However, the literature shows they obtain good results.

2.2.4 Shape features

Other researchers try to use shapes to detect an object. In [Opelt et al., 2006], the authors propose to use boundary of object to detect them. Actually the exact boundary of any object is not simple to extract, so object's boundary is represented as a set of shape fragments. They also propose a method to select the fragments as well as a boosting algorithm adapted to shapes. Very similar features can be found in [Wu et Nevatia, 2005] where edgelet is introduced to discriminate human body parts using their silhouette. In [Sabzmeydani et Mori, 2007], shapelets are introduced and are used to detect pedestrian from the image. Shapelets is extracted from regions with low level gradients. The main benefit of the features is that

they try both to segment and detect the object in the image. However, they are very sensible to occlusion and illumination conditions.

2.2.5 Template based features

This designates features where templates of facial characteristics are used. Here, a template of a specific anatomic part (e.g. eye, nose, mouth) is designed. Generally, this template contains shapes and models possible deformation. Then, candidates are extracted and compared with the template. In [Jian et Honglian, 2009], eyes are detected using a multi-angle template. Candidates are extracted using morphological operators and the symmetrical characteristics of the eyes. In [Yuille et al., 1989], a deformable template is used. Here, template deforms itself by minimizing a cost function to find the best fit. The main drawback of these methods is the difficulty to generalize the templates under various illumination and scale conditions. For example, an eye template will fit candidate to template only if eye is open. The main benefit of these approaches is they are able to process sub-face part.

2.2.6 Knowledge based features

The last set gathers together features which include knowledge and spatial information of the face. In [Gizatdinova et Surakka, 2007], twenty landmarks, for example the eye or mouth endpoints, are detected automatically on images with different expressions, illumination conditions. Spatial knowledge is introduced to the method to improve accuracy of these landmarks. In [Kotropoulos et Pitas, 1997], a face detection method is proposed using mosaic images. To achieve detection, this method requires rules related to the spatial information of a face. Actually knowledge based features are hybrid features. Face knowledge is often used to fix some limitations and to remove aberrant values of other features.

In general, many methods in face detection use local features. Since the amount of local features is often too large, they use to either select the most representative features or create a subspace of these features. In most of the cases, a further step using machine learning techniques allow face detection or pose estimation.

Landmarks or control points are widely used in face alignment issues, but rarely for face detection purpose. They are difficult to generalize and thus depend on the learning data. Moreover, they are not discriminating enough to separate face and non-face images. However, when a face is

detected and we are sure of its presence in an image, they give impressive visual results. Therefore, they are good candidates to detect the positions and sizes of face salient anatomic elements (eyes, nose and mouth). However, we will show in the next section that despite impressive visual results, control points are not as accurate as they seem to be and thus why have been motivated to choose to detect more approximately these elements.

2.3 Motivation

As we said in the state of art, there are many types of face features. Some of them are local, related to shape, edge orientation or texture information. Other methods try to find control or salient points. Others try to extract higher level information. In the case of local features, the main drawback is the absolute necessity of a labeled database. Of course, it is possible to use Haar filters, LBP, HoG to detect different salient parts of a face. However, the training step needs tens of thousands of positive and negative examples. All must be correctly scaled and positioned.

The first idea we may have, for example to estimate the head pose is to reconstruct the face from different face samples of a video sequence. Many face recognition systems rely on a face 3D model such as in [Chu et al., 2014]. So, first, we must localize the positions of equivalent points in all frames. This task has been already achieved in many approaches relying on the detection of anatomic salient points. Many methods require a machine learning steps using these points which are labeled manually by human vision, but it seems that localizing these points is quite difficult.

We made an experimentation to show that human beings are not able to localize these points reliably. To verify the difficulty in using these anatomic points, we test the robustness of the positions of these points. For this purpose, we try to make a 3D reconstruction from equivalent points in a video sequence with good illumination conditions. First, let us choose the control points of the face. As shown in Figure 2.5, seventeen control points are labeled.

- Each eye is labeled with 4 points: left (A), right (B), up (C) and down (D) end points.
- Nose is represented by 5 points: nose tip (E), nose upper central point (F), left (G) and right (H) end points and nose basis central point (I).
- Mouth has also 4 control points: the central upper (J) and lower points (K), left (L) and right (M) end points

We use a video of a single person. Each frame contains the same face. Throughout this video, the head turns to the right and left. All faces of this video sequence were labeled manually with the control points defined above.

These are the steps of the protocol for the experiment:

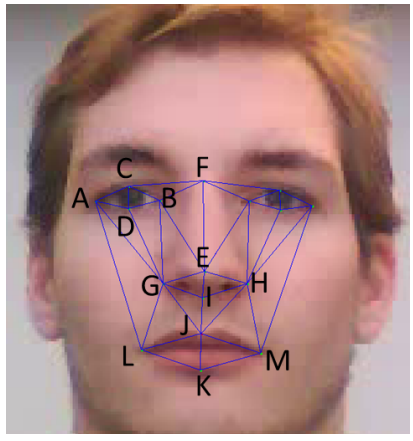

Figure 2.5: Face on a frame of a video sequence as well as its related control points.

1. First, we estimate the fundamental matrix.
2. Then, using equivalent points of a pair of face samples, the 3D reconstruction gives the 3D relative positions of these points.
3. Finally, we try to measure the reliability of these points. If these points are credible, it means that human vision can localize these anatomic points reliably, otherwise it means that human vision is not able to localize these points with accuracy.

In order to obtain the 3D relative positions from two sets of control points, we must first get the fundamental matrix. Given the focal lengths f_x and f_y, as well as the projection coordinates (c_x, c_y) of the image center , we are able to compute the fundamental matrix F as shown in formula 2.8.

$$F = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.8}$$

All we have, is sets of homologous control point coordinates. In theory, if x_i and x_j are two homologous points, they must respect equation 2.9.

$$x_i^t F x_j = 0 \tag{2.9}$$

However, these values are rarely equal to zero. The aim is to find the matrix F which minimizes $x_i^t F x_j$. Note that 7 pairs at least of homologous points are needed to compute the fundamental matrix. We used

the RANSAC algorithm ([Chum et Matas, 2008]). RANSAC is especially efficient when outliers must be excluded.

1. Among all control points, 7 points of the first image and their equivalent points in the second image are randomly chosen.
2. The fundamental matrix is then computed with the chosen 7 pairs of points.
3. For every homologous pair of points (except the 7 pairs initially chosen), if $x_i^t F x_j \leq \epsilon$ then we consider that the pair x_i and x_j respects equation 2.9.
4. We count the number of pairs which respect $x_i^t F x_j \leq \epsilon$.
5. Then, we restart step 1 to 4.
6. Finally, we choose the matrix F which has the highest number of matching points.

Once the fundamental matrix F is computed, we used Hartley triangulation method ([Hartley et Sturm, 1995]). The triangulation gives three types of results.

- When the equivalent sets of control points are almost the same, in other words when face doesn't move a lot between two frames, the triangulation gives 3D coplanar points (Figure 2.6).
- When the equivalent sets of control points change a lot, for example when the first set is extracted in a frontal view whereas the second is extracted for an almost profile view, the triangulation gives 3D linear points.
- When the equivalent sets of control points change a little, the triangulation gives a correct 3D representation (Figure 2.7).

When triangulation fails, it gives coplanar or linear 3D points. Given $X_i = (x, y, z)_t$ the i-th result 3D point of the triangulation, given $A = (X_1 X_2 ... X_n)$ the matrix of all 3D points, we compute the variance-covariance matrix AA^t of size 3×3, then we extract the 3 eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$, then eigenvalues are normalized with λ_1. The 3 normalized eigenvalues $\lambda_{n1} > \lambda_{n2} > \lambda_{n3}$ are given by equation 2.10.

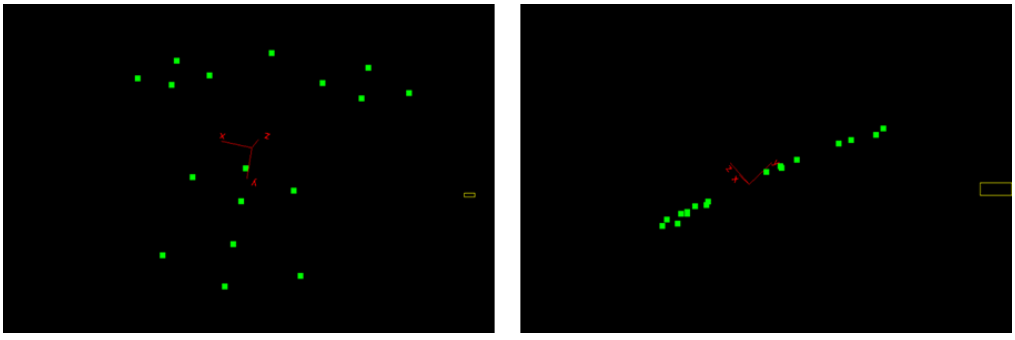

Figure 2.6: Coplanar triangulation: left and right images are two different projections of the same 3D result.

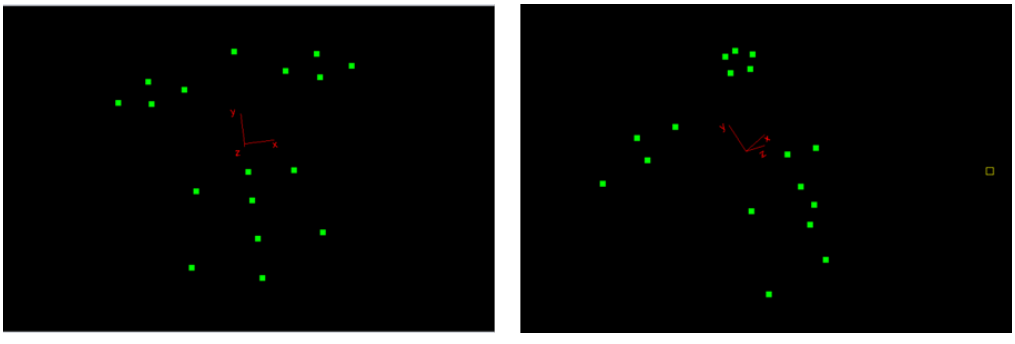

Figure 2.7: Correct triangulation: left and right images are two different views of the same result.

$$\begin{cases} \lambda_{n1} &= 1 \\ \lambda_{n2} &= \frac{\lambda_2}{\lambda_1} \\ \lambda_{n3} &= \frac{\lambda_3}{\lambda_1} \end{cases} \tag{2.10}$$

Table 2.1 shows the normalized eigenvalues according to the distribution of 3D reconstruction points. As expected, in the coplanar distribution, the lowest eigenvalue is negligible compared to the other whereas the two lowest eigenvalues are negligible compared to the highest one in the linear case. It means that a reconstruction fails if the lowest normalized eigenvalue λ_{n3} is negligible compared to 1. A threshold $t_{norm} = 0.01$ is enough to know if the 3D reconstruction succeeds. As we can see in table 2.1, when reconstruction fails, λ_{n3} is far from reaching t_{norm}.

Most of the time, the triangulation fails. First, this experiment shows that it is not so simple for human beings to label the control points correctly. For our vision, many different point positions seem correct. It means that for human vision, a point in a frame has a large number of candidate

Table 2.1: Examples of normalized eigenvalues according to 3D reconstruction point distribution.

Distribution	λ_{n1}	λ_{n2}	λ_{n3}
Coplanar	1	0.74	0.0009
Linear	1	0.028	0.00055
Correct triangulation	1	0.83	0.18

homologous points in another frame. Moreover, experiment shows that human vision can better localize control points when face view is frontal. Indeed, most of successful triangulation includes a frontal view. The more profile the view is, the more difficult localizing accurately control points is.

Thus, human vision does not require an accurate localization of face elements to detect face. Indeed, even if we were not able to localize equivalent control points accurately, we are able to know that the object of this video is a frontal view. Therefore, we work towards approximate determination of face elements and not towards anatomic points.

2.4 The face salient element extraction method

In this section, we will present how we extract facial salient elements. We suppose face detection succeeds, face is detected in a square face window of length L pixels. It means that we know the order of magnitude of the face scale. In a first part, we discuss the properties of Haar-like features before we propose the general overview of the proposed method. Then, the different steps are described.

2.4.1 Energy from Haar-like features

As we said in the previous state of the art, Haar-like features are simple and powerful tools in particular in face detection. The computation of any value associated with a given Haar-like pattern is done in constant time. Indeed, the integral image or summed table allows the computation of the sum of the intensity in a rectangular window in only 4 accesses in the summed table. In other words, the feature values computation does no longer depend on the width and height of the features.

Another benefit of Haar-like features is the infinite number of possible patterns. However, all patterns do not have the same importance. To be robust, all methods which use Haar-like features must select carefully the set of patterns they want to apply. Haar-like features are versatile features. Patterns with small size will describe local texture. With bigger size, they will describe lines, edges and contours. Finally, when Haar-like feature size is large enough, the descriptor will describe more high level structure. In this part, we will introduce the features we use, how to compute energies and how to combine them as well as the sense of these features.

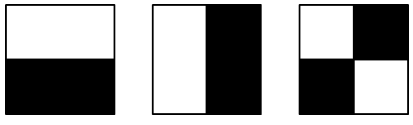

Figure 2.8: Studied Haar Patterns.

As shown in Figure 2.8, we consider the horizontal, vertical and diagonal patterns with different sizes. The horizontal pattern is sensitive to contours of horizontal direction whereas the vertical one is sensitive to vertical direction. The diagonal pattern is not sensitive to diagonal direction. It is more a pattern which describes this specific form.

Given $H(w, h)$ the Haar-like filter of width w and height h and (X, Y) the position of the central pixel, the associated value is given by 2.11.

$$f_{H(w,h)}(X,Y) = \sum_{(x_1,y_1)\in white(X,Y)} I(x_1,y_1) - \sum_{(x_2,y_2)\in black(X,Y)} I(x_2,y_2) \quad (2.11)$$

We define the energy map E_H related to a given Haar-like filter H as the absolute of f_H (definition 2.12).

$$E_{H(w,h)}(X,Y) = \left| f_{H(w,h)}(X,Y) \right| \quad (2.12)$$

Here, an energy map depends on the pattern and on the size of the Haar filter. First, let us see how the size affects the horizontal Haar pattern on a face example (Figure 2.9). The example face has a size of 364×364.

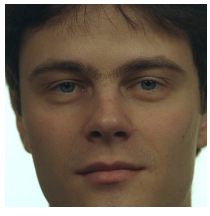

Figure 2.9: The face example used in this section.

2.4.1.1 Energy with Haar horizontal patterns

Figure 2.10 shows the energy maps of the horizontal pattern of Haar filters according to the width w and the height h of the filter. First, let us consider how the size of the filter affects the energy. As it is difficult to evaluate directly with the energy maps, a threshold is applied manually to each map to get a suitable binarization. The figure 2.11 shows the energy maps on which a suitable threshold has been applied. The chosen threshold must give a binary image where salient face elements (eyes, nose basis and mouth) are highlighted whereas other parts like face boundaries or other noise are not taken into account.

It should be recalled that the aim here is to extract salient anatomic part of the face: left and right eyes, nose tip and mouth. From now on, we will focus on extracting these areas. A few remarks should help us to achieve our task.

- When, at least, one of the width w or height h of Haar horizontal filter is low, energy maps are sensitive to noise, especially when $w = 4$ and $h = 4$.

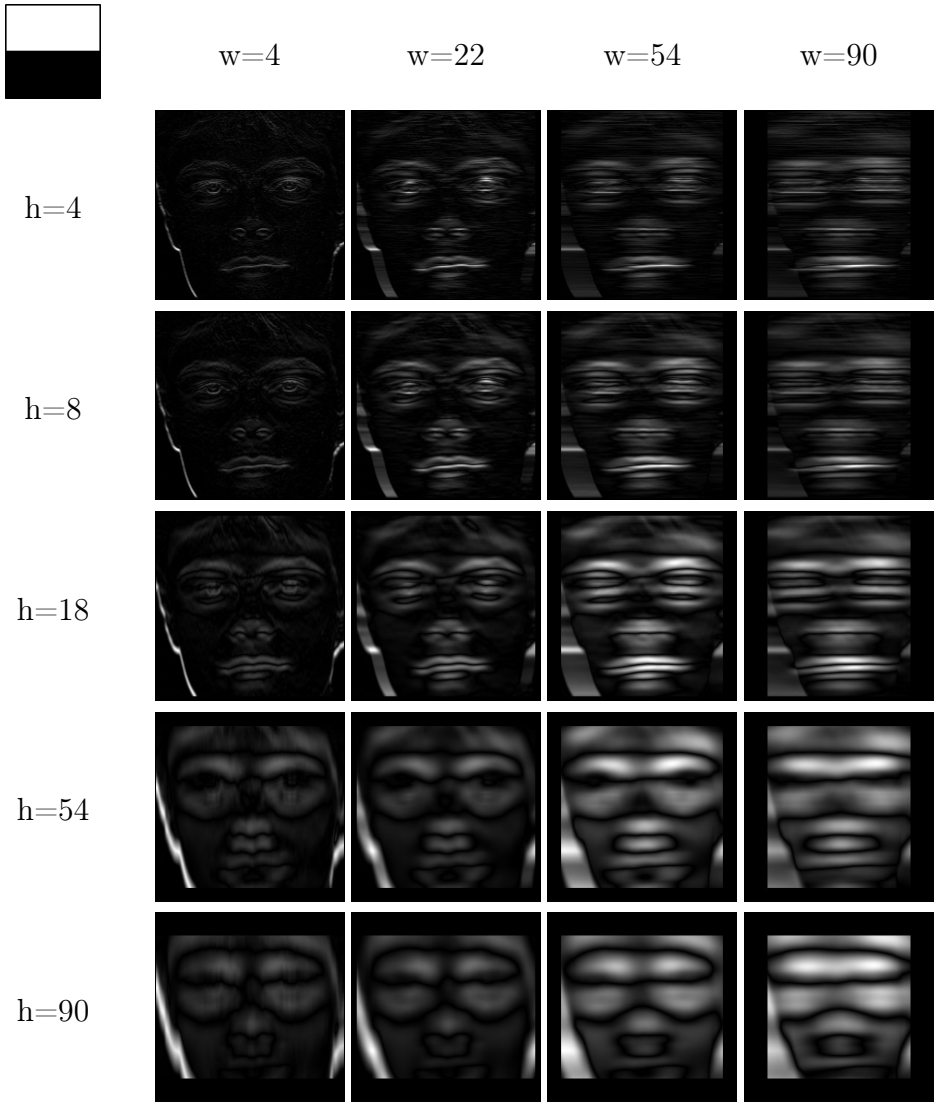

Figure 2.10: Energies of horizontal Haar filter according to the filter width w and height h in pixels. The face window width and height equal both to 364 pixels

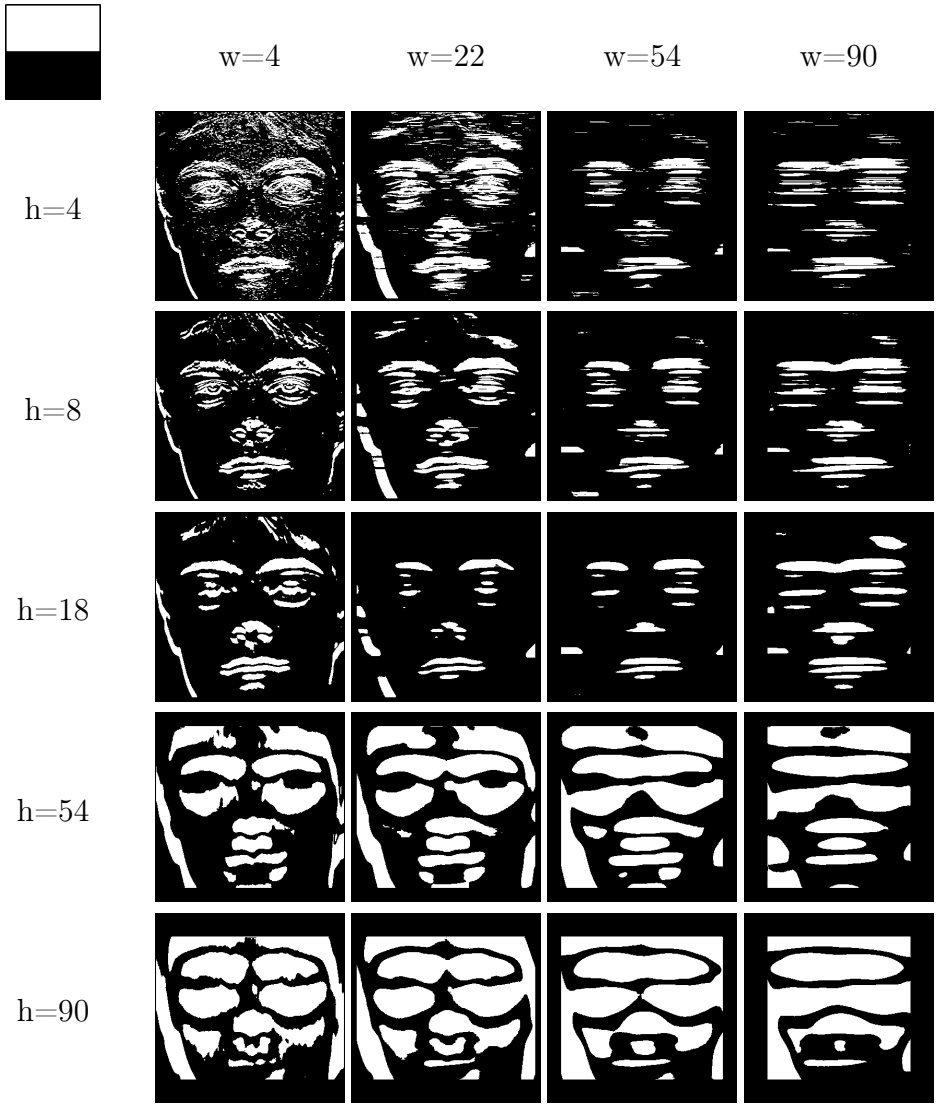

Figure 2.11: Binarized horizontal energy maps. Thresholds are chosen manually with a visually suitable value to illustrate.

- The horizontal Haar energy map should have a better response with area which contains mainly horizontal edges. However, when $w = 4$ or $w = 22$, the Figure 2.11 shows that vertical or oblique lines at the border of the face have a great horizontal energy. When Haar filter has a small width, the filter is too local. Therefore, except for strictly vertical lines, contours which are not exactly vertical have a non negligible energy. On the contrary, when width w of Haar filter is high enough, in our example when $w \geq 54$, lines with almost vertical direction disappear.

- When the width w or the height h of the pattern is too high, regions tend to merge, other regions, such as hair or forehead have also a great energy. Almost all the face has a great energy.

As a consequence of all these remarks, for this example, suitable filter width and height are $w = 54$ and $h = 18$. First, horizontal filter must have a higher width than height. Indeed, a higher width prevents the high horizontal energy value on edges with approximate vertical direction. A ratio of $1/3$ seems to be a good compromise. Moreover, remember that the face is a 364×364 pixels image. So, as we can see in the Figure 2.9, an eye has almost an approximate width of $1/6$ of the square face example size L. So eyes have a width of 60 pixels. Notice that nose tips and mouths have the same order of magnitude in terms of width, although nose tip width is generally lower than eye width and mouth width is generally higher. Then, a good width for horizontal Haar filter is the width of the eye, since all salient anatomic elements that we want to find have the same order of magnitude.

2.4.1.2 Energy with Haar vertical patterns

In a similar way, we now study the energy associated with a Haar vertical pattern. In this Figure 2.12, the energy is high on the boundary of the face. In order to better appreciate the results, a threshold is applied on Haar vertical energy map, presented in Figure 2.13. Here too, we can make a few remarks about the behavior of the vertical energy maps according to the width and height of the Haar patterns.

- Like the horizontal energy maps, when the width and height of the vertical pattern are too low, the map is too noisy.
- When the width is too high, even though it is a vertical pattern, the vertical energy is not local enough: although there are still approximate vertical lines near the boundary of the face, these lines are too wide (they look more like regions than like lines).
- The same observation can be made with a high height. The lines are not accurate enough.

In this example, both vertical Haar patterns $H(8, 54)$ and $H(18, 54)$ seem to be the most suitable ones. With these vertical patterns, salient vertical lines are highlighted and noise is reduced. They are the most representative. Moreover, we can see that vertical energy is globally high

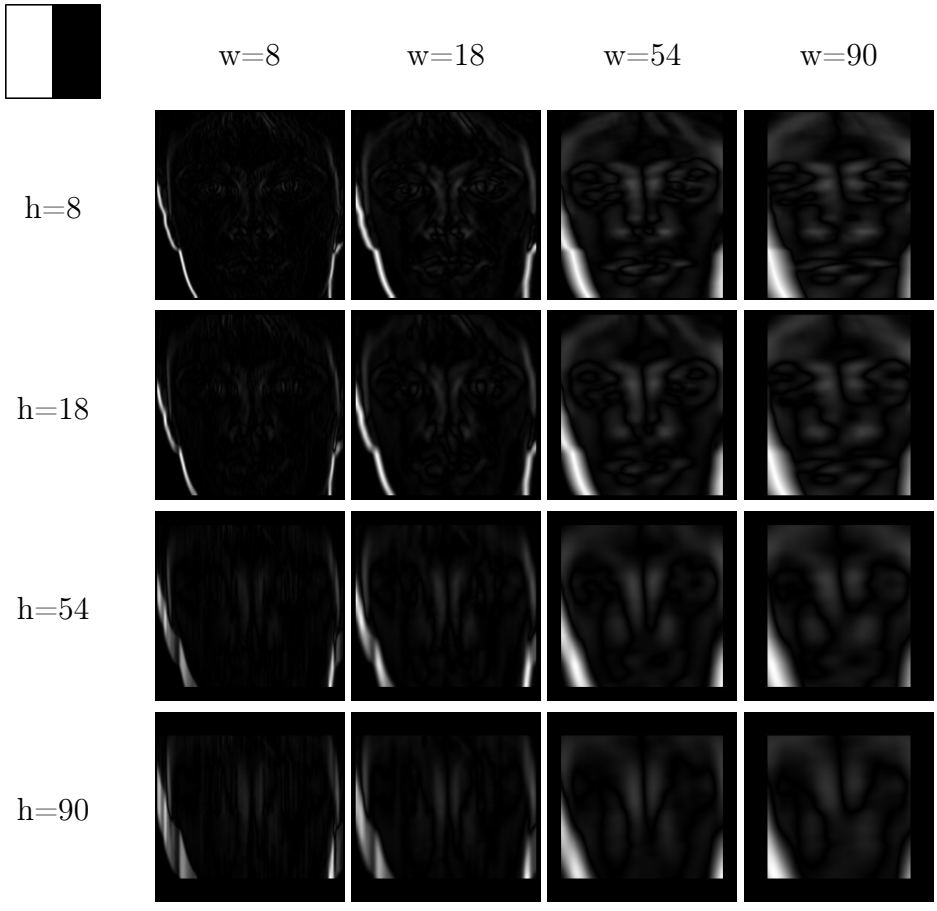

Figure 2.12: Normalized energy maps of vertical Haar filter according to the filter width w and height h in pixels. The face window width and height equal both to 364 pixels.

on the boundary of the face. Remember that the aim is the extraction of the face salient areas (eyes, nose tip and mouth). Although the vertical maps do not highlight the face salient elements, they may be useful to estimate face contours.

2.4.1.3 Energy with Haar diagonal patterns

First, as a reminder, the diagonal Haar pattern is not specifically sensitive to line with an approximate diagonal direction. It is more a pattern able to respond, to see the local stability, in terms of intensity of an area. As we can see in Figure 2.14, a strictly vertical or horizontal line has a low energy when the pattern is a square ($w = h$). With a small size, the pattern will be sensitive to atypical area or noise. This pattern is rather interesting when

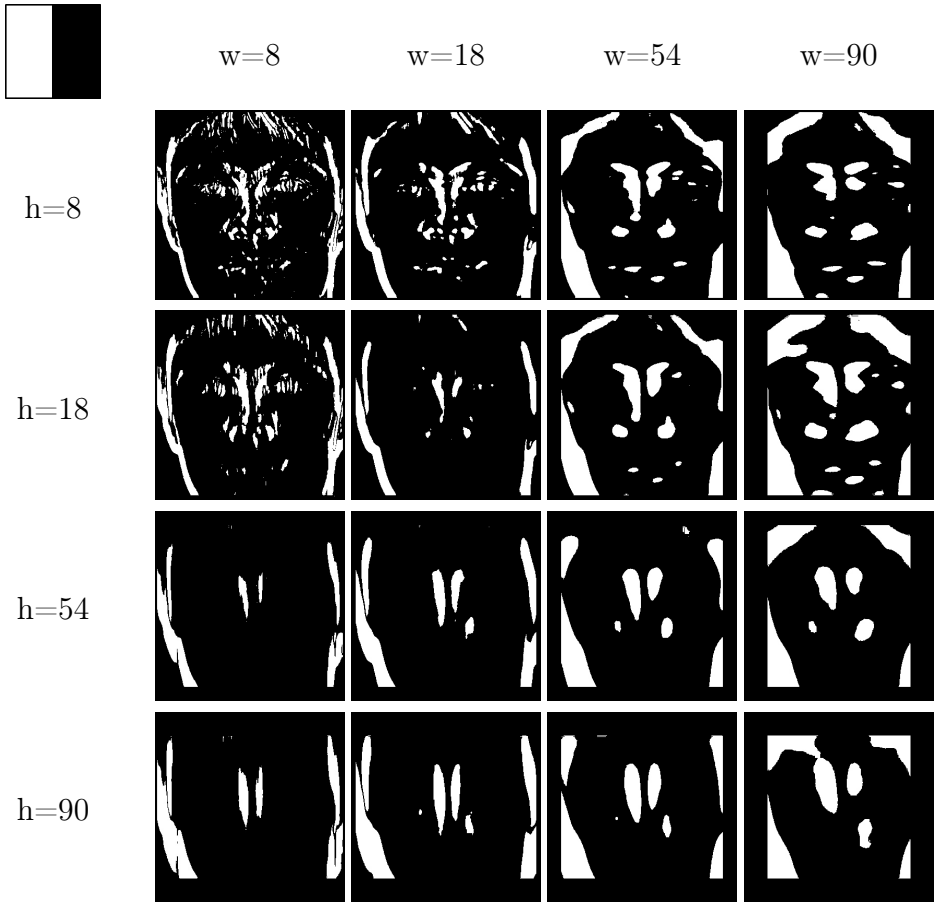

Figure 2.13: Binarized vertical energy maps. Thresholds are chosen manually with a visually suitable value to illustrate.

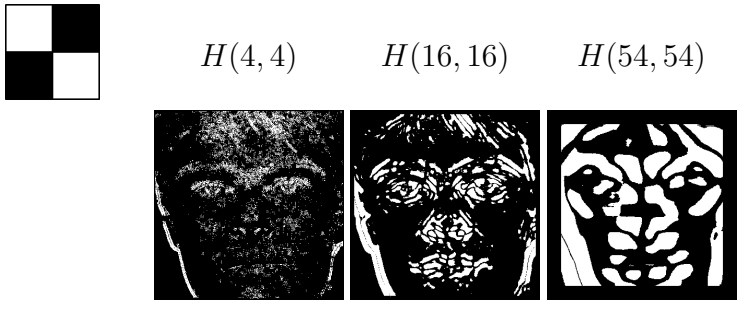

Figure 2.14: Binarized diagonal energy maps. Thresholds are chosen manually with a visually suitable value to illustrate.

we want to see areas where intensities are constant. For this task, local minimums are more representative than local maximums. For example, in Figure 2.14, when the $w = 4$ and $h = 4$, the results are very noisy, in

particular because of small squares created by jpeg compression. However, when the pattern size is higher, for example when $w = 54$ and $h = 54$, the energy map has low values when the area around the pixel is locally stable or symmetric. This is the reason why eyes, nose and mouth are often separated vertically showing the symmetry of these face elements. One can think this a good way to detect face symmetry. However, as the face turns, face elements are no longer symmetrical. So, the face symmetry detection will fail. As we can see, this pattern will not be efficient in face element detection.

2.4.1.4 Combination of horizontal and vertical energy

Here we propose to combine the information given by several energy maps. We define two operators.

Linear combination The first combination of a Haar horizontal pattern Hh and vertical pattern Hv is a weighted combination $H\alpha$ of these Haar energies (equation 2.13).

$$\begin{cases} E_(Hh, Hv, X, Y) &= \alpha_h \left| f_{Hh}(X,Y) \right| + \alpha_v \left| f_{Hv}(X,Y) \right| \\ \alpha_h + \alpha_v &= 1 \end{cases} \tag{2.13}$$

The weights α_h and α_v make the pattern areas comparable, since their size may differ. For example, let's take Hh the horizontal pattern of size $(54, 18)$ and Hv the vertical pattern of size $(18, 54)$. Since the size of both pattern are the same, $\alpha_v = \alpha_h = 0.5$. The result will be an edge detector. Figure 2.15 shows some results with different sizes of patterns.

Figure 2.15 shows that the lower the size of Haar filter is, the noisier the edges are. So, when the size increases, the amount of edges is reduced and only the most representative ones are detected. In any case, in such representation of edges, the notion of direction is lost, only intensities are taken into account.

Non-linear combination However, the direction information is relevant. In order to keep it, another approach combining both horizontal and vertical energies is also proposed. Let us take an horizontal pattern Hh of size $w = a, h = b$ and its analogue vertical pattern Hv of size $w = b, h = a$. Let us define three sets, the set S_{hori} contains points the neighborhood of which contains an approximate horizontal direction line,

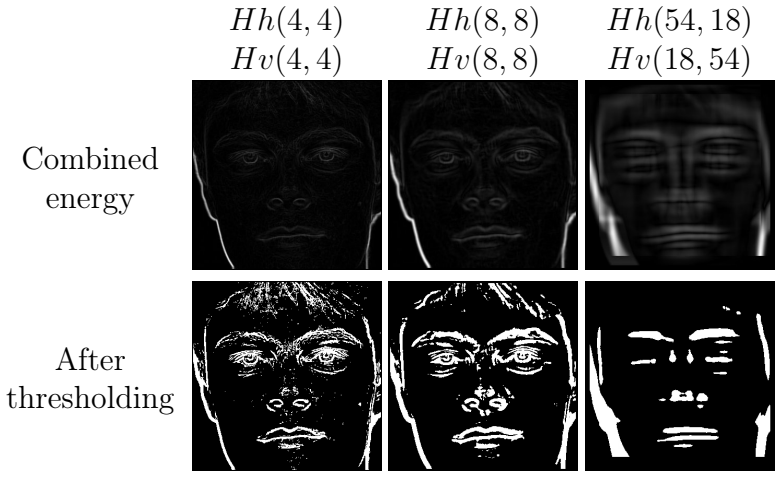

Figure 2.15: Linear combination of horizontal and vertical Haar energies. the result is an edge detector.

the set S_{verti} contains points the neighborhood of which contains an approximate horizontal vertical line. Finally, the set S_{other} contains points of other elements(background, neither vertical nor horizontal line).

For a given position (X, Y) in the horizontal and vertical energies, respectively $E_{Hh}(X, Y)$ and $E_{Hv}(X, Y)$ are computed. If $E_{Hh} > E_{Hv}$, then we have at position (X, Y) a line which is more "horizontal" than "vertical". Similarly, if $E_{Hv} > E_{Hh}$, then we have at position (X, Y) a line which is more "vertical" than "horizontal". Moreover, if both energy values at position (X, Y) are low, it means that these energies are negligible compared to the highest ones. For example, an almost constant area will give low values of horizontal and vertical energies. We introduce two thresholds:

- the first threshold $C_d > 1$ defines the ratio from which a line is considered as vertical or horizontal.

- the second threshold $0 < C_v < 1$ defines the value from which an energy is no longer negligible.

The formula 2.14 shows 4 definitions, the first two conditions D_{hori} and D_{verti} are related to local direction and the two last conditions V_{hori} and V_{verti} are related to the minimum magnitude of the energies needed to be taken into account.

$$\begin{cases} D_{hori}(X,Y) &= \frac{E_{Hh}(X,Y)}{E_{Hv}(X,Y)} > C_d \\ D_{verti}(X,Y) &= \frac{E_{Hv}(X,Y)}{E_{Hh}(X,Y)} > C_d \\ V_{hori}(X,Y) &= E_{Hh}(X,Y) > max\,(E_{Hh}) \times C_v \\ V_{verti}(X,Y) &= E_{Hv}(X,Y) > max\,(E_{Hv}) \times C_v \end{cases} \tag{2.14}$$

With these 4 definitions, we are able to define some clusters of pixels using some logical combinations as shown in the equation 2.15.

$$\begin{cases} S_{hori} &= \{(X,Y)/D_{hori}(X,Y)\ and\ V_{hori}(X,Y)\} \\ S_{verti} &= \{(X,Y)/D_{verti}(X,Y)\ and\ V_{verti}(X,Y)\} \\ S_{other} &= \{(X,Y)/\text{ all other combinations}\} \end{cases} \tag{2.15}$$

Figure 2.16 shows some results according to the two thresholds C_d and C_v, and according to the size of the horizontal Haar pattern Hh and the vertical one Hv.

- Green values visualize the set S_{hori} of lines with an approximate horizontal direction.
- Red values visualize the set S_{verti} of lines with an approximate vertical direction.
- Other pixels of set S_{other} are represented in blue.

As we can see, when size of both patterns are low, the response of non-linear combination is, once again, sensitive to noise. As the size of patterns grows, the sets are better separated. However, when the ratio between a given energy value and the maximal energy value is less or equal to 0.1 ($C_v \leq 0.1$), the response is too strong, especially for the largest pattern of the examples.

2.4.2 Overview of face anatomic elements extraction

As we mentioned earlier, face salient anatomic parts are more visible with horizontal energy map. Many researches try to extract anatomic part independently. Some try to extract eyes, whereas others try to find mouth from face image. These methods are able to find a specific part, but don't consider the whole face. However, it is quite obvious that the eyes, nose and mouth in a face are related. Positions of these elements, their spatial

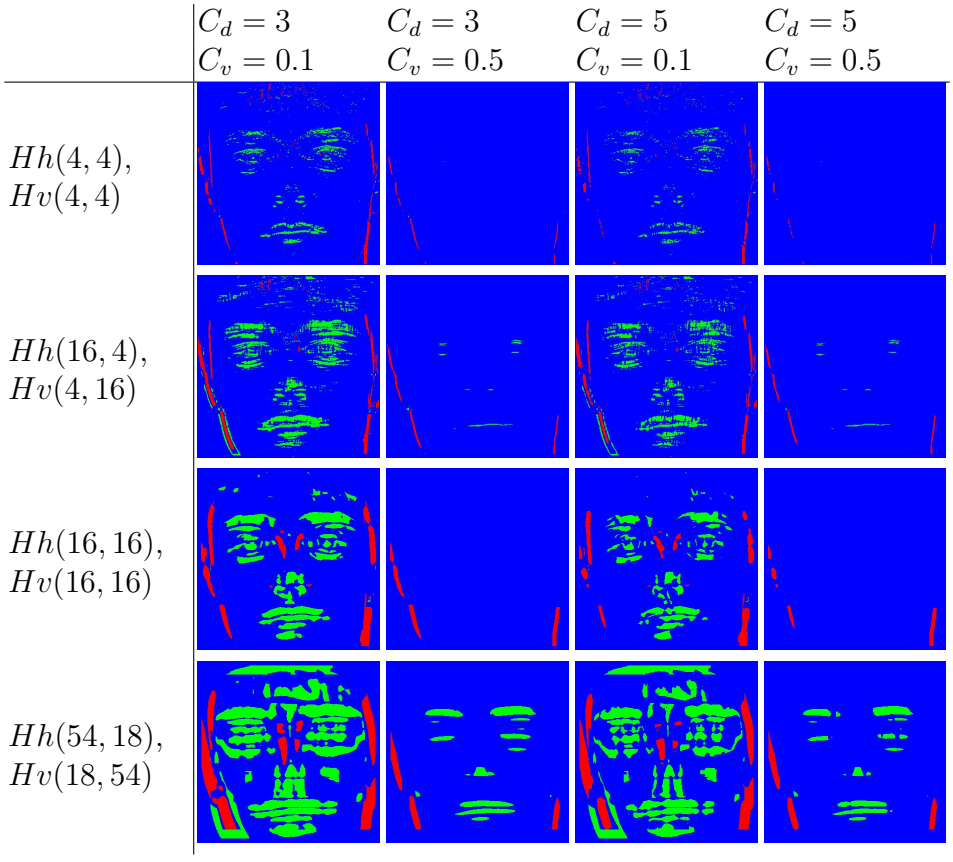

Figure 2.16: Visual results of non-linear combination of horizontal and vertical energies. Values which belong to S_{hori} is represented in green, values of S_verti is represented in red and other values of the combination (belonging to S_{other}) in blue.

relationship are crucial information which can improve their detection from a face.

Besides, some control point localization method shows a great performance and achieve good detection of some control points of the face. Almost all of them are designed to find these control points when face contains both eyes. In other words, these methods must localize both eyes even when one of them is not visible. Many researchers may think this is an improvement, since, for example, these methods are able to localize approximately the iris control points when eyes are closed. In such cases, the results are visually impressive. However, in a context of a video sequence, profile views of a face often appear. Since these methods need to place all control points, it will still localize the hidden eye, even when it is obviously not visible. Control points localization methods are powerful in controlled context, for example, in the context of a camera in front of a driver in

a car, or in a context of facial expression analysis where face images are frontal. In real world video, these methods are not robust enough to roll rotation.

Moreover, we have seen that localizing manually these control points is more difficult than it seems. Actually, when human beings try to localize for example the endpoints of an eye, they will localize them at different positions. So it is more exact to say from a human visual point of view, that these endpoints are better defined by regions including these endpoints.

Our face salient element extraction method do not try to extract landmarks but try to localize the eye, the nose and the mouth regions globally using a prior information of human face structure. In this part, we try to extract the eyes, nose and mouth bounding boxes of these facial elements. Our method is not designed to detect both eyes. If only one is visible, because of an occlusion or because the face image is a profile view, it will detect only one eye.

Before we extract the face salient regions, we introduce a preprocessing step which should make the extraction easier. Indeed, the face window contains obviously the face but also a background which could delude the method we propose.

Therefore, globally, our method is composed of two main steps as it is presented in Figure 2.17.

- First, vertical energy map is used to extract vertical borders of the faces. The result is a mask enabling to exclude the background.
- Second, horizontal energy map is used to extract salient face anatomic parts. The result is the bounding boxes of eyes, nose and mouth.

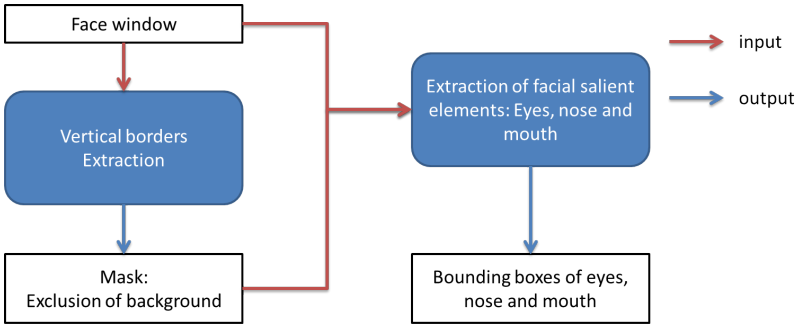

Figure 2.17: Global view of face salient anatomic element extraction.

The first part will be studied in part 2.4.3. Figure 2.18 shows the tasks which achieve face vertical border extraction.

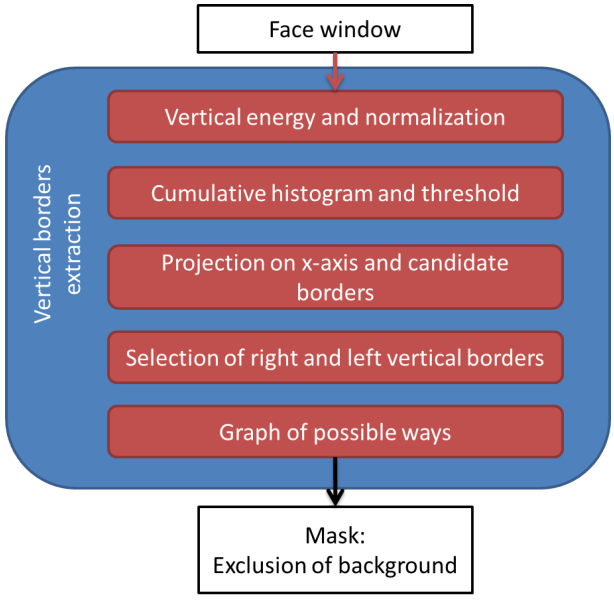

Figure 2.18: Scheme of vertical face borders extraction, the result is a mask which separates the face from the background.

The second step is the extraction of salient anatomic elements. Bounding boxes of eye, nose and mouth regions are extracted. This extraction is divided in four main parts. There are two inputs in this face anatomic part extraction step. First, there is the face window extracted from Viola and Jones face detector or any other face extractor, second is the mask which gives the search area.

- Task 1: a horizontal energy map is computed from the horizontal Haar pattern.
- Task 2: candidate anatomic regions are extracted according to each binarization threshold.
- Task 3: from all candidate anatomic regions, a multi-threshold analysis is applied to extract suitable regions of eyes, nose and mouth.
- Task 4: a validation is made on horizontal Haar pattern size. If it is validated, then we keep the selected regions of the previous task, otherwise horizontal Haar pattern size is modified and the process restarts at task 1.

Figure 2.19 shows a scheme of the face salient elements extraction step.

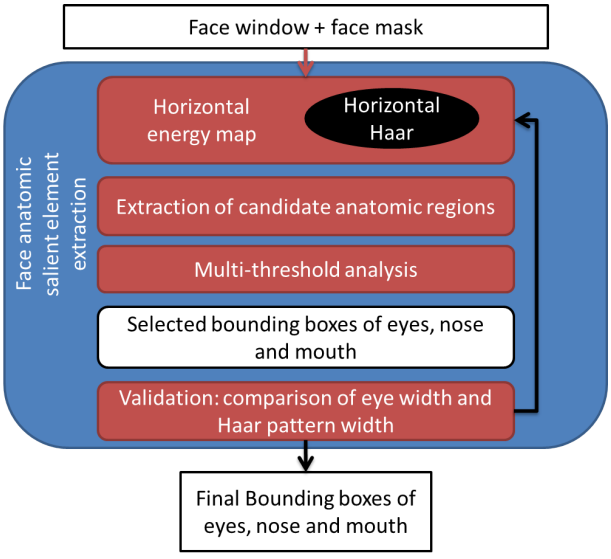

Figure 2.19: Extraction of face salient elements extraction.

2.4.3 Extraction of face vertical contours

As we saw in the previous section, a face is mostly delimited on its left and right sides by its vertical borders. A first step consists in extracting borders of approximate vertical direction. Such task may improve the detection of salient face elements. We see several benefits of this approach.

- First, finding the left and right borders reduces the search area.
- Then, separating the background from the face enables to suppress lines outside the face. Since our approach is based on energies from Haar-like features, some approximate horizontal and vertical lines may appear in the background, it is useful to know where these lines have to be ignored to extract salient areas of a face.

In order to extract the vertical contours of a face using Haar energies, we must first fix some parameters.

- First, we have to choose a pattern. Since vertical contours must be extracted, we will simply choose the vertical Haar-like patterns.
- Second, we have to fix the size of the Haar pattern to extract the vertical energy of the face image.
- Finally, we must choose a suitable threshold from which a vertical energy of the map can be considered as sufficient to contain in its neighborhood a line of approximate vertical direction.

This will be explained in sections 2.4.3.1 and 2.4.3.2. Once the size of the vertical pattern has been fixed and once the vertical energy map has been computed, we are able to extract and select a mask of the face, as explained in sections 2.4.3.3 and 2.4.3.4.

2.4.3.1 Chosen vertical Haar pattern and its associated energy map

As we said in part 2.4.1.2, vertical Haar pattern can be a good candidate tool to extract contours of a face. As a reminder, we assume face detection succeeded in a square window of size L pixels. Since the order of magnitude of the face scale is given by L, the size of vertical Haar filter is simply a ratio of the face window length L. However, as we said, when the size is too small, vertical energy is sensitive to noise, or to small local maximums. Besides, when it is too high, the filter is no longer local. The width w_{Hv} and height h_{Hv}, inspired by face mean anatomy, are given by formula 2.16.

$$\begin{cases} w_{Hv} &= 0.05 \cdot L \\ h_{Hv} &= 0.15 \cdot L \end{cases} \tag{2.16}$$

Figure 2.20 shows 4 face windows and their associated vertical energy maps. As expected, most of the pixels with highest energy are located on the face left and right boundaries. Since the pattern and its size have been chosen, we can now compute the energy E_{Hv} given by formula 2.17.

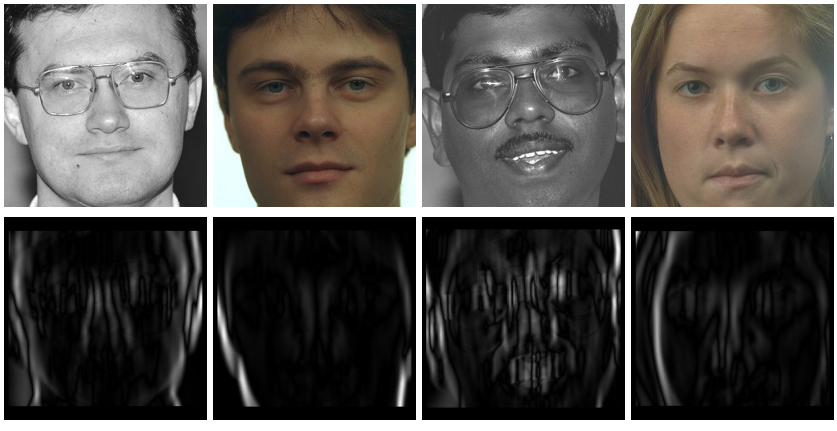

Figure 2.20: The first line contains the original images, whereas the second one contains normalized vertical energy maps.

$$
\begin{aligned}
f_{Hv}(X,Y) &= \sum_{(x_1,y_1)\in white(Hv)} I(x_1,y_1) - \sum_{(x_2,y_2)\in black(Hv)} I(x_2,y_2) \quad (2.17)\\
E_{Hv}(X,Y) &= |f_{Hv}(X,Y)|
\end{aligned}
$$

Then, the energy is normalized (En_{Hv}) using the maximum value of E_{Hv} in the face window I (equation 2.18).

$$
\begin{aligned}
M_{Hv} &= \max_{(x,y)\in I} E_{Hv}(x,y) \quad (2.18)\\
En_{Hv}(X,Y) &= \frac{E_{Hv}(X,Y)}{M_{Hv}}
\end{aligned}
$$

Figure 2.20 shows some examples of vertical normalized energy computed from four faces.

2.4.3.2 Adaptive threshold on vertical Energy map

Once the energy map has been computed, to define a binarised version, a threshold has to be chosen: energies which exceed this threshold will indicate the pixels with a neighborhood of vertical approximate direction. Figure 2.21 shows some binarized vertical energy maps using different values of threshold.

- Globally, for a given threshold t, the binarized vertical energy map does not have the same response. For example, in Figure 2.21, with $t = 0.059$, this threshold may be suitable to detect contours of face 'b'. However, it is not the case for face 'c' where binarized vertical energy map is too noisy.
- Binarized energy map better discriminates vertical contours when background intensities are very different from those of the face. We can see it for image 'a', 'b' and the left part of image 'd'.
- As the face pixel intensities are almost the same as the background ones, the binarized vertical energy map detects contours of the face, but also other parts, such as nose sides, parts of glasses, eyes...

From these remarks, it is quite obvious that a fixed threshold will not be suitable to find vertical contours of a face. However, pixels with high

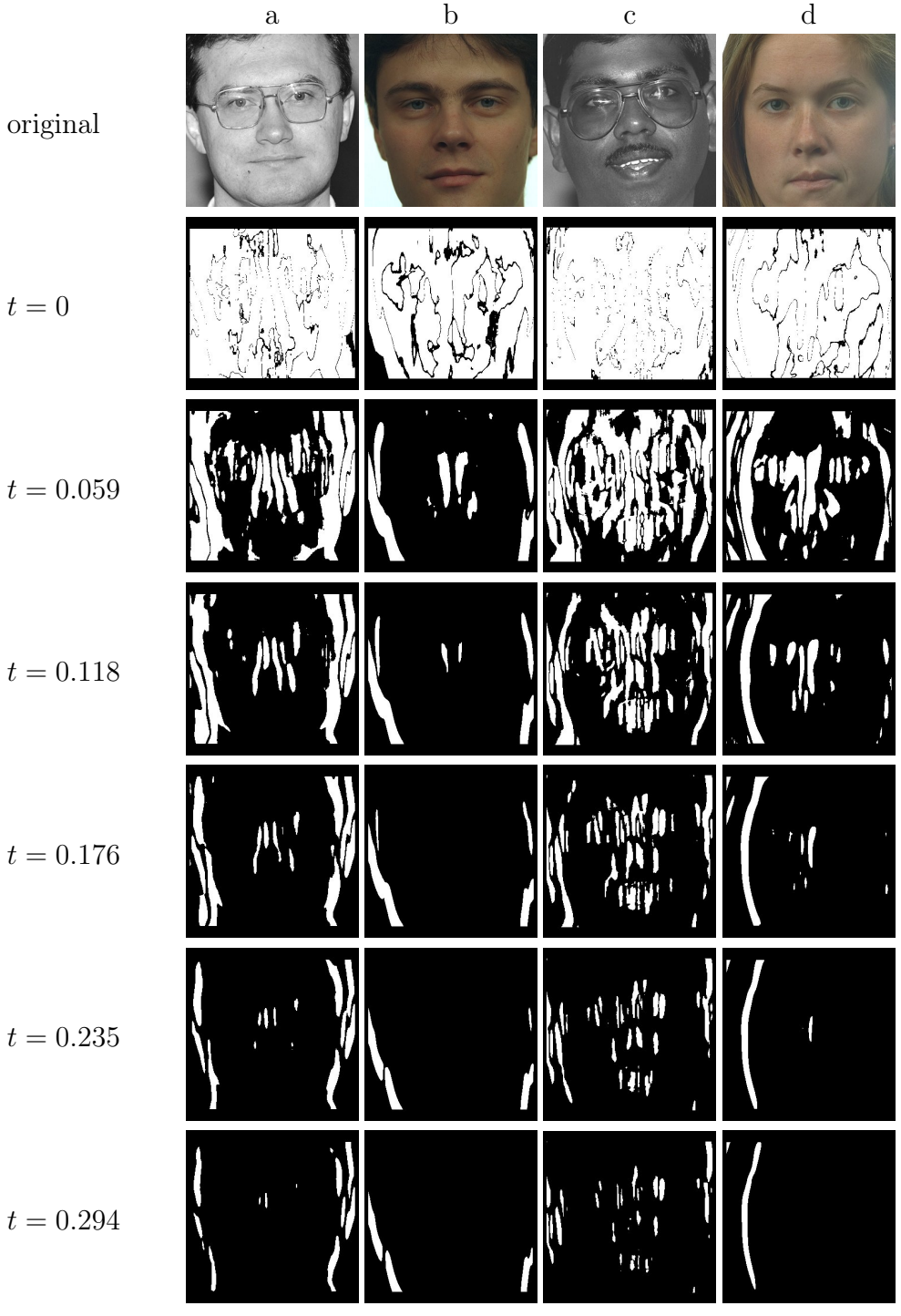

Figure 2.21: The first line contains the original images, The others contain binarized vertical energy maps according to the threshold t.

vertical energy are located on the vertical boundaries or next to nose. When background pixels have almost the same vertical energy as those of the face (example 'c'), elements of nose and the borders have pixels of highest vertical energy. When background pixels intensities are very different from those of the face, borders are the elements with high vertical energy. In order to compute a proper binarization threshold, histogram and cumulative histogram (Figure 2.22) of the vertical energy map are computed.

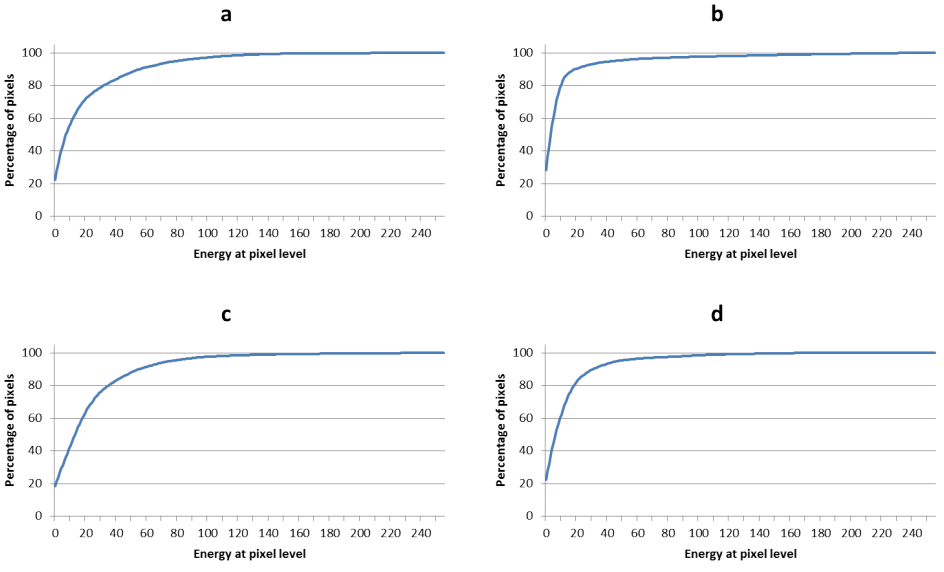

Figure 2.22: Cumulative histograms in percentage of vertical energy map. Original images are those of Figure 2.21.

As we can see in the cumulative histogram 'b' of Figure 2.22, when the background pixel intensities are very different from those of the face, the cumulative histogram increases rapidly; in other words, high vertical energies are well separated from the others. Although cumulative histogram 'd' increases more slowly than 'b', it still increases faster than 'a' and 'c'.

Histograms 'a' and 'c' increases more slowly; it means that there are more pixels with intermediate energies, since the background intensities are closer to those from face. We are interested in finding pixels with high values, since there are more pixels with intermediate energies, finding a suitable threshold in such cases is more difficult. However, for all face windows, boundaries and elements close to nose are the parts which have highest energies. We propose to consider all the highest 15% energy values, since we saw that we had more stable and less noisy binarization of vertical energy map. Given $C_+(x)$ the cumulative histogram, and S the area of the

face window, the binarization threshold t_V^* is given by formula 2.19.

$$t_V^* = \min_{C_+(t)>0.85\cdot S} t \tag{2.19}$$

Table 2.2: Adaptive threshold obtained on images of Figure 2.21.

images	a	b	c	d
t_V^*	0.176	0.051	0.173	0.090

Table 2.2 shows the computed thresholds associated with the 4 images of Figure 2.21. As we said before for images 'b' and 'd', the more different the intensities of the background and the face are, the lower the adaptive threshold is. The value of adaptive threshold can also be seen as a confidence measure of robustness in our vertical border detector. Indeed, when the value of the threshold is low, it means high energy values are well separated from the other. In other words, there are less intermediate vertical energies. Figure 2.23 shows the binarized energy maps of 4 images.

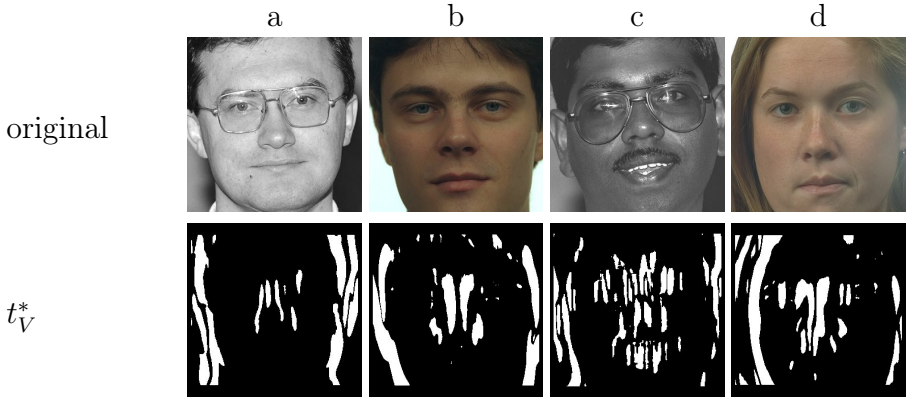

Figure 2.23: Binarized vertical energy map using adaptive threshold.

As we can see in Figure 2.23, binarization with this adaptive threshold seems more interesting than with a fixed threshold. The results are to be compared with binary images of Figure 2.21. Indeed, for all images, high vertical energies are located on face vertical borders and on elements close to the nose. Within these binarized images, face borders are now to be extracted.

2.4.3.3 Extraction of candidate borders

The borders will be materialized by selected connected components of binary images. In each connected component, at each pixel, the vertical energy is greater than t_V^*. It appears in the map as line of approximate vertical direction. Since largest components should be taken into account (Figure 2.24), small vertical connected components, the area of which is less than 1% of the connected component with the greatest area are removed. As lines have almost the same direction, the previous computations are done on the bounding boxes of connected components to make the process faster. In order to extract face vertical borders, candidate vertical borders are generated. Each candidate border is then defined by vertical connected components which have a common projection on the x-axis. So we generate a graph giving the number of pixels belonging to the connected component bounding boxes for each image column.

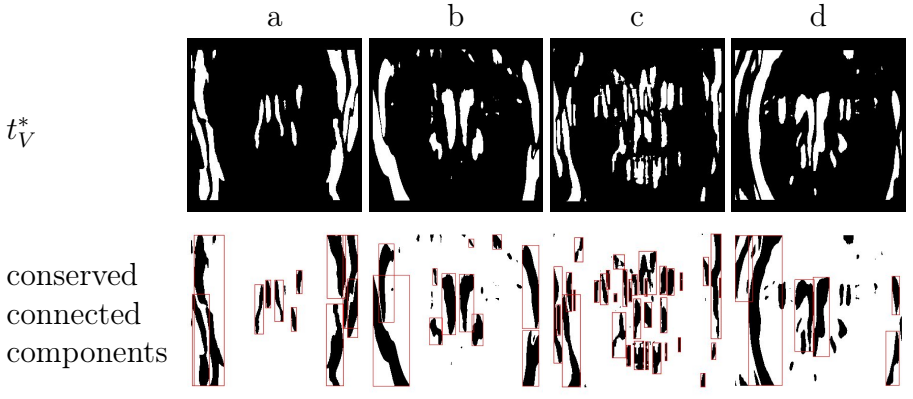

Figure 2.24: Connected component bounding boxes the area of which is more than 1% of the area of the greatest bounding box.

Figure 2.25 shows the number of pixels belonging to connected component bounding boxes according to the image column. All connected components which have a common projection on x-axis (i.e. which contribute on the same vertical) are merged to form a candidate border. If a candidate border B_i is separated from another one B_j by less than 1% of the face window width, both are merged to generate a single candidate border. Given two candidate borders B_i and B_j, the merging criterion is given by formula 2.20.

$$\begin{aligned} B_k &= B_i \cup B_j \text{ if } \min_{(x_1,x_2)} |x_1 - x_2| < 0.01 \cdot L \\ &\quad \text{with } x_1 \in proj_x(B_i) \text{ and } x_2 \in proj_x(B_j) \end{aligned} \tag{2.20}$$

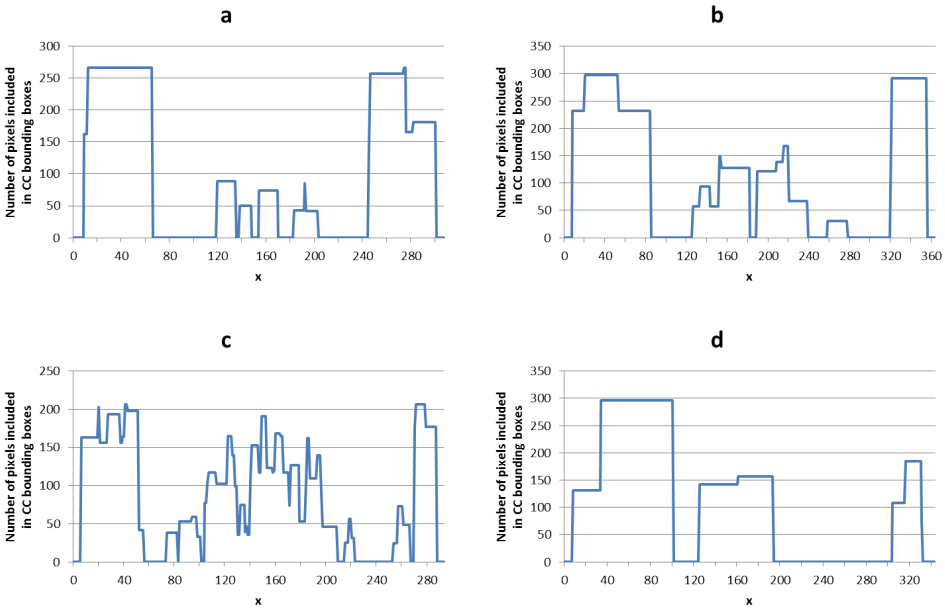

Figure 2.25: Number of pixels belonging to connected component bounding boxes according to the image column.

Then, each candidate border B_i is defined by the set S^i_{CC} of connected components with the maximum cumulative height H_i, the lowest abscissa is denoted as X^i_{left} and the highest one by X^i_{right}.

Let also mention some basic remarks.

- A candidate border must be completely either on the left side or the right side of the face window. In other words, if a candidate border is in the center part of the face, it is not significant.
- Each of the left and right side of the face can have at most one vertical border.
- Each of the left and right face vertical border must be the most salient candidate borders.

2.4.3.4 Selection of candidate borders and construction of the face mask

This section will show how both left and right vertical borders can be selected and how the face mask is then built. When we speak about the left view, we consider the point of view of the camera. So, the left vertical

border, will be at the right side in the point of view of the face. As we said, a candidate border cannot be in the center part of the face window. Let us define three sets: S_{left} will contains all candidate vertical borders of the left side of face window whereas S_{right} will contain all candidate vertical borders of the right side of face window. The set S_{center} will contain all candidate vertical borders in the center part of face window. Given B_i a candidate border, sets S_{left} and S_{right} are defined by formula 2.21.

$$\begin{aligned} B_i \in S_{left} \quad &, \text{if } X^i_{left} \leq \tfrac{1}{2} \cdot L \text{ and } X^i_{right} \leq \tfrac{1}{2} \cdot L \\ B_i \in S_{right} \quad &, \text{if } X^i_{left} \geq \tfrac{1}{2} \cdot L \text{ and } X^i_{right} \geq \tfrac{1}{2} \cdot L \\ B_i \in S_{center} \quad &, \text{if } X^i_{left} < \tfrac{1}{2} \cdot L \text{ and } X^i_{right} > \tfrac{1}{2} \cdot L \end{aligned} \tag{2.21}$$

At this stage, we know that a candidate border which belongs to S_{center} will not be selected.

Each side of the face can have at most one vertical border. Let us consider the left side. The face border must also be the most salient. We define B_{left} the selected vertical left border as the highest border of S_{left} which also must be higher than all candidate borders of S_{center} as expressed in formula 2.22. H_i designates the height of the bounding box of B_i.

$$\begin{aligned} B_{left} = B_i \quad &, \text{with} \quad B_i \in S_{left} \\ &\text{and} \quad H_i = \max_{B_j \in S_{left}} H_j \\ &\text{and} \quad H_i > H_k, \forall B_k \in S_{center} \end{aligned} \tag{2.22}$$

The right border B_{right} can be selected similarly by the formula 2.23.

$$\begin{aligned} B_{right} = B_i \quad &, \text{with} \quad B_i \in S_{right} \\ &\text{and} \quad H_i = \max_{B_j \in S_{right}} H_j \\ &\text{and} \quad H_i > H_k, \forall B_k \in S_{center} \end{aligned} \tag{2.23}$$

The second condition in 2.22 and in 2.23 selects the highest vertical border among respectively the sets S_{left} and S_{right}. The last condition in 2.22 and in 2.23 selects a vertical border, only if the border is greater than candidate vertical borders of the center. Sometimes, a face border is not or is only partially in the face window. In theses cases, only a little part of the border is visible in the face window and thus, the third condition

in 2.22 and in 2.23 is often no longer satisfied. Nevertheless, we add this last condition because it should give a better precision but a worse recall. Remember the aim is to reduce the search area of anatomic salient parts. It is better to find less vertical borders than generate false face borders.

Figure 2.26 shows the selected vertical left border (green rectangles), as well as vertical right border (red rectangles).

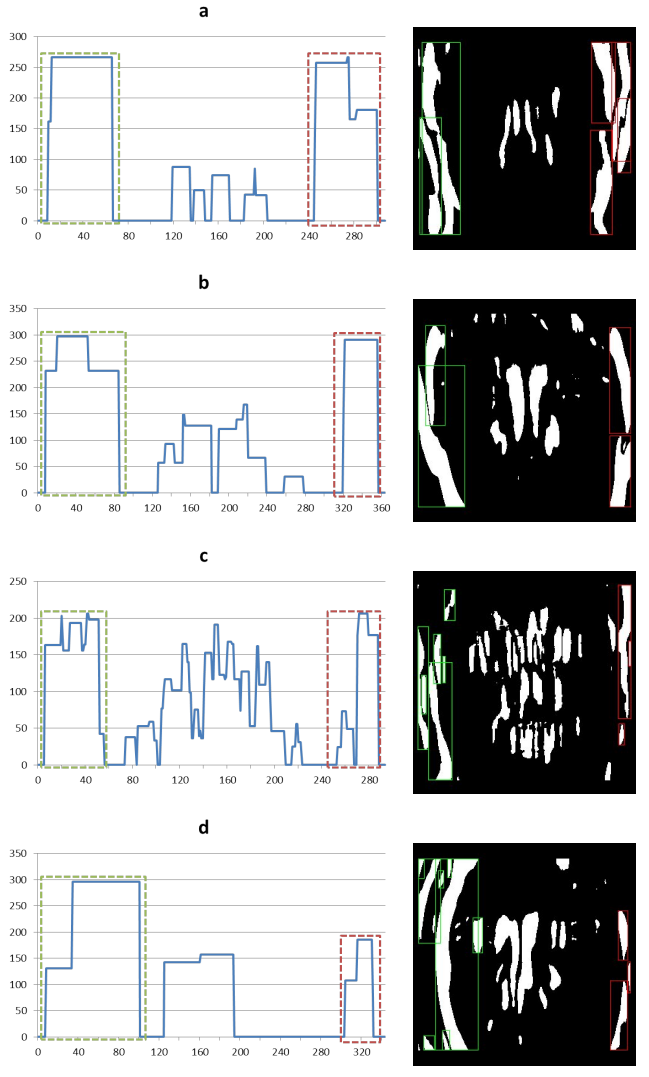

Figure 2.26: Selected left and right vertical borders; the left vertical border is represented in green boxes and right one is represented in red boxes.

Now that vertical borders are selected, mask can be generated. However, the selected vertical borders is composed of several vertical connected components and sometimes some of them are at the same height, we have

to choose among the connected components of a given selected border those which will be considered as the face vertical border.

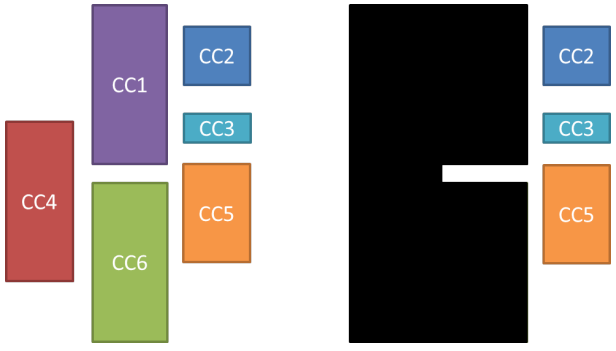

Figure 2.27: Schematic example of a selected left vertical border; the left side of this figure shows the connected components of this border, the right side shows the mask.

As shown in the Figure 2.27, the connected component CC1 is at the "same height" as the connected component CC2. In order to make a hierarchy of all these connected components, we generate an oriented graph. An oriented edge from the vertex CC_i to the vertex CC_j, it means that CC_i is "immediately above" CC_j.

As we can see, the graph should represent the relative positions according to connected component ordinates and heights. Indeed, a part of CC1 is above CC2, but a part of CC1 is also below CC2. Hence, it it difficult to say which one is relatively above the other. Concerning the components CC1 and CC4, a part of CC1 is above CC4, however, there is no part of CC1 which is below CC4 and thus we can say that CC1 is above CC4. In other words, CC1 and CC2 are at the same level contrary to CC1 and CC4. Although we can say that CC2 is above CC3, we consider that both CC2 and CC3 are at the same level as CC1, otherwise the graph will be inconsistent, since it will depend on the first vertex used to create the graph. Simirlarly, CC4 and CC5 are at the same level, whereas CC4 or CC5 are also above CC6. Moreover, there is no edge from CC1 to CC6 because CC4 and CC5 are above CC6 and hence, CC1 is not immediately above CC6.

To build the graph, we need to know the conditions which gather some connected component in a same level. Let consider a group of connected components which are at the same level, another connected component is also at this level if its (horizontal) projection on y-axis is included or includes the projection on y-axis of, at least, one connected component of this level.

In other words, Given CC_i a connected component, P_y is the projection on y-axis operator. We define a level $\mathcal{L}_k$ starting from $\mathcal{L}_1$. $\mathcal{L}_1$ contains the highest connected component CC_1 (with the lowest ordinate). Then,

$$CC_i \in \mathcal{L}_1 \Leftrightarrow P_y(CC_i) \subset P_y(CC_1)$$

Iteratively, $\mathcal{L}_{k+1}$ is defined. It contains the connected component CC_l with the highest position in $\mathcal{L} \setminus \bigcup_{i=1}^{k} \mathcal{L}_k$ and

$$CC_i \in \mathcal{L}_{k+1} \Leftrightarrow P_y(CC_i) \subset P_y(CC_l)$$

The graph is recursively built. The left part of Figure 2.28 shows the final graph generated from this example.

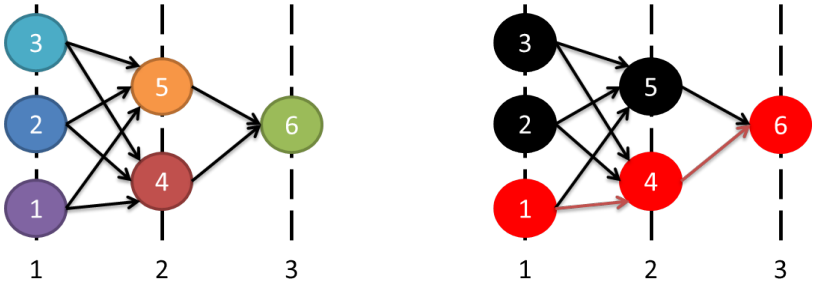

Figure 2.28: Generated graph and selected connected components from the example of Figure 2.27.

As shown in the right part of Figure 2.28, at each level of the graph, since it is the most significant in this level, the connected component with the greatest height is chosen. Finally the components CC1, CC4 and CC6 are selected. Figure 2.29 shows the mask generated from our initial 4 examples.

2.4.4 Eyes, nose tip and mouth extraction

Now that we have a mask which is able to separate face area and the background in the face window, it is time to focus on the salient face elements extraction. These areas are left and right eyes, nose tip, mouth and they will be defined as rectangular regions or bounding boxes. This method was described in [Pyun et al., 2014a].

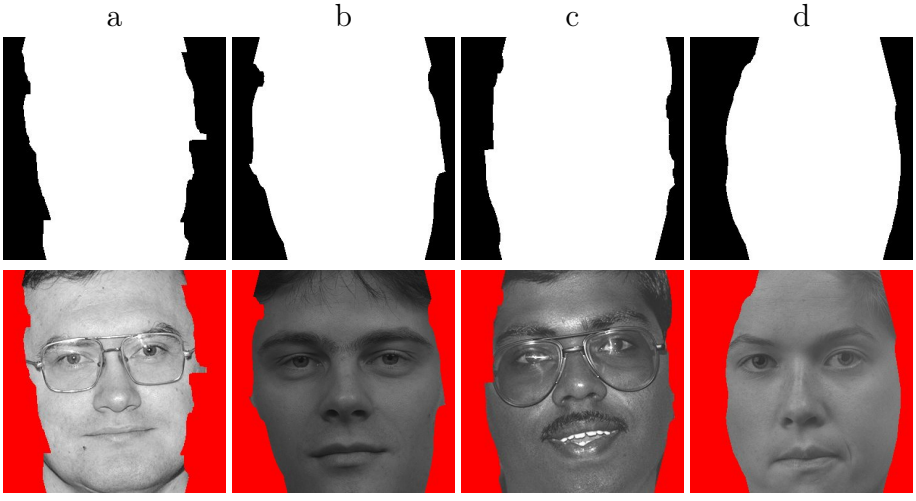

Figure 2.29: Vertical border face masks.

2.4.4.1 Adaptive horizontal Haar pattern optimization

As we said, salient elements of the face, such as eyes, nose and mouth have, most of time, an approximate horizontal direction. The horizontal Haar pattern (Figure 2.30) is a good candidate to find these regions. However, even if we know which Haar pattern will have the best response in terms of face anatomic element detection, we still do not know what a suitable size for this pattern is. If the size is too small, the energy map generated from this pattern will be noisy, whereas if the size is too large, this Haar filter will not be local enough compared to the face scale. We assume face detection succeeds; the whole face is detected in a face window. Two faces in their respective window will have approximately the same scale compared to the scale of the whole window.

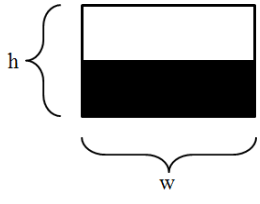

Figure 2.30: Haar horizontal mirror pattern of width w and height h.

As a reminder (see section 2.4.1.1), we assume that a suitable width of horizontal Haar filter is the width of an eye. With such a width, it is a good compromise between an energy less sensitive to noise and an enough local filter. The height of the horizontal Haar filter is fixed to 2/3 of the width. In order to make the size adaptive, we propose an iterative process

to fix the width of the Haar pattern. w and h are respectively the width and the height of the Haar filter. R and L are respectively the bounding boxes of right and left eyes.

- Initialization: First, the width of the filter is initialized with a value of a sixth of the whole face window width, $w = \frac{1}{6} \cdot W$.
- The method will extract bounding boxes of salient face anatomic regions. So, we will get the bounding boxes R and L of right and left eyes.
- Let us call w_{mean} the mean of left and right eye widths; $w_{mean} = 0.5 \cdot w_R + 0.5 \cdot w_L$. if $0.2 \cdot w_{mean} < w < 1.2 \cdot w_{mean}$ then we will keep all the selected regions as the final result of our method, otherwise we modify w and h values, ($w = w_{mean}$, $h = \frac{2}{3} \cdot w$) and consider to process extraction again with the horizontal Haar filter of modified size.

The evolution of Haar filter width is defined by equation 2.24.

$$\begin{aligned} w_{mean} &= \frac{w_R + w_L}{2} \\ \text{if } |w_{mean} - w| &< \epsilon_{haar} \text{ then} \\ w = w_{mean} &\quad \text{and} \quad h = \frac{2}{3} \cdot w \end{aligned} \tag{2.24}$$

Here, $\epsilon_{haar} = 0.05 \cdot w_{mean}$. As a result, the Haar filter will have almost the same width as the eyes width. Eye width reference was preferred to those of nose or mouth for two reasons. First, the eyes are the most significant parts of a face. We assume that there is globally more information in eye regions than in the others. Second, although eyes, nose and mouth have a size of a same order of magnitude, a single eye region is generally larger than nose tip region but smaller than the mouth. So, a Haar filter with the width of an eye will have a size close to any face element size.

2.4.4.2 Extraction of candidate anatomic regions

Now that we know which Haar pattern to use and its size, we are able to compute the energy map E_{Hh} using the chosen horizontal Haar filter in the equation 2.11.

Then, the normalized energy En_{Hh} is computed using the maximum value of E_{Hh} in the face window I (equation 2.25).

$$M_{Hh} = \max_{(x,y)\in I} E_{Hh}(x,y) \tag{2.25}$$
$$En_{Hh}(X,Y) = 1 - \frac{E_{Hh}(X,Y)}{M_{Hh}}$$

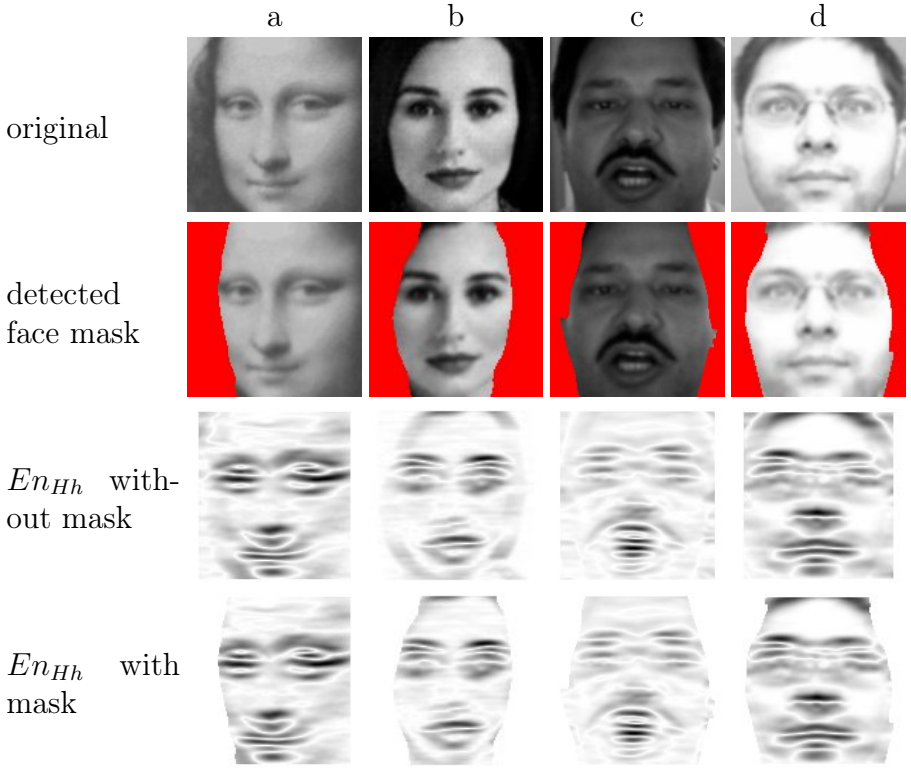

Figure 2.31: From top to bottom: original face window, detected face mask, normalized horizontal energy map of the original image, normalized horizontal energy map after applying the face mask.

In the equation 2.25, the normalized energy is a value between 0 and 1. However, contrary to the initial value of the energy E_{Hh}, the lower a value in the normalized energy map is, the more confident in the presence of an approximate horizontal direction line in the neighborhood we are. The normalized energy map can be seen as an inverse energy map where values are between 0 and 1. So, compared to the initial energy map, values are normalized and inverted. The inversion makes the method easier to understand and to formalize.

Figure 2.31 shows some examples of normalized horizontal energy map with or without the face mask defined in section 2.4.3. With this nor-

malization, points with lower energy are more stable in terms of having a horizontal direction in its neighborhood.

Threshold discussion At this moment, horizontal Haar filter size is fixed. A normalized horizontal energy map En_{Hh} is then computed with this Haar filter. However, a threshold must be applied to this energy map. Above this threshold t, values are considered as non significant. Under this threshold, values of energy are kept, since they represent pixel with a neighborhood of approximate horizontal direction. Figure 2.32 shows some examples of binarization with different thresholds t using or not the face mask. As we can see, the face mask is particularly useful with high threshold values. It removes most of the parasite information of the background.

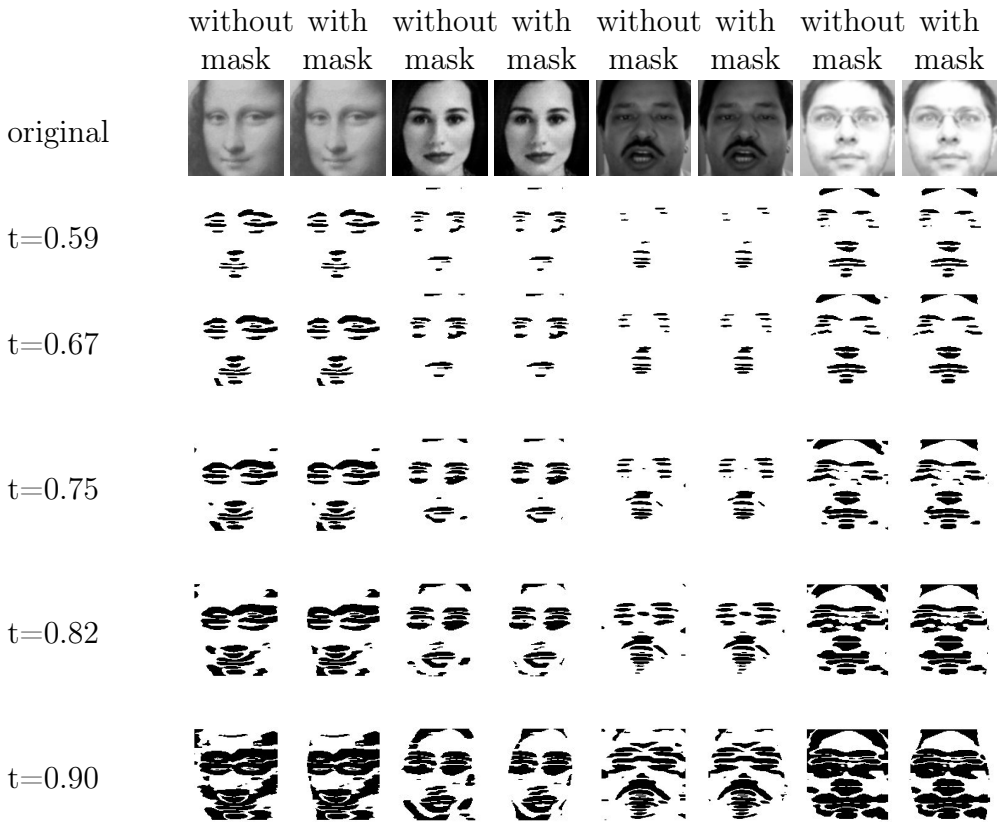

Figure 2.32: Examples of energy map En_{Hh} binarization according to threshold t without the face mask and after applying the face mask.

As expected a fixed binarization threshold is not suitable for all images. For example a threshold of 0.59 seems to be suitable for the image 'a' of Figure 2.32, whereas it is not the case for images 'b', 'c' and 'd'.

It seems that energy map binarization depends on face illumination conditions. When the illumination applied on face creates more contrast, a lower threshold seems to be enough, whereas when there is less contrast between the skin and the other anatomic parts of the face, a greater threshold is needed. Moreover, in a single face, illumination condition can vary from an anatomic element to another. For example in the face window 'b', eyes and mouth are visible with a threshold value of 0.67 or 0.75. However, the nose tip is only visible for a threshold of 0.75. If a threshold of 0.75 is suitable for eyes and mouth, a threshold of 0.82 is needed to detect nose tip accurately. So, a suitable threshold for an anatomic element is not necessarily a suitable one for the others.

These remarks show that we need to find a suitable threshold for each anatomic salient region of the face. On the other hand, several binarizations are able to extract a given anatomic region. It means that a threshold which is able to extract this anatomic region is not unique. So, the aim is to choose a suitable threshold among the possible suitable thresholds. As we cannot know which threshold is a suitable one for a given anatomic region, we first decide to extract candidate anatomic regions according to all fixed thresholds. In other words, for each threshold and binarization, the method will extract eyes, nose tip and mouth as candidate. A further step will decide, for each anatomic region of the face which threshold is the most suitable one.

Candidate extraction In order to extract candidate anatomic regions, the binarization threshold is fixed. Then, connected components, more accurately the bounding boxes of connected components, are extracted. Since face salient elements are in the same order of magnitude, it also means that too small connected components can be ignored. All connected components the bounding box area of which is less than 1% of the largest connected component bounding box area are removed. Figure 2.33 shows the bounding boxes of the connected components according to the thresholds.

When the threshold increases, components tend to merge compare to those with lower thresholds. With high threshold values, some connected components gather together several anatomic regions. So, for such cases, it is not possible to separate these anatomic regions. Working with the bounding boxes makes the process faster. At this moment, the aim is to gather together connected component bounding boxes of the same anatomic region and also to remove connected components which belong neither to eyes, nor to nose tip, nor to mouth. Remark that with this normalization, as the threshold increases, areas taken into account increase

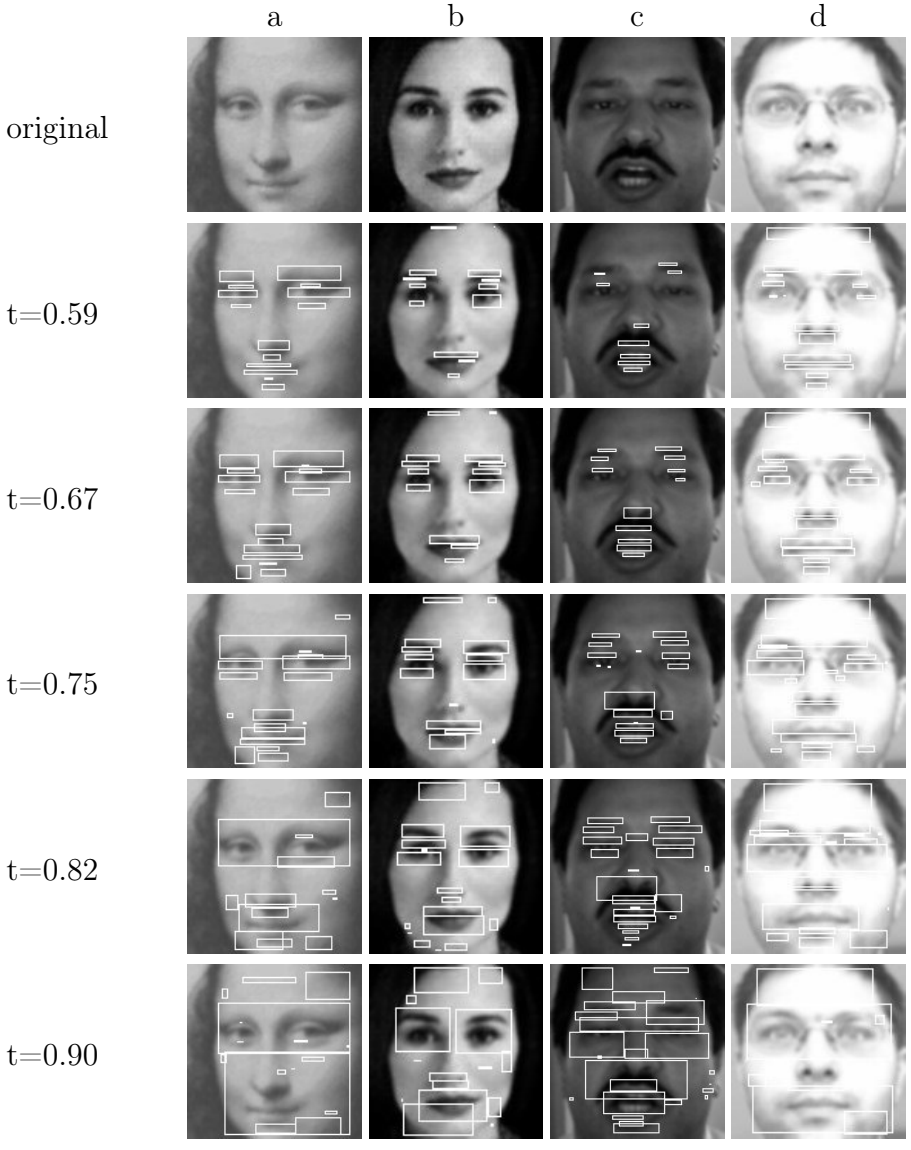

Figure 2.33: Bounding boxes of connected components according to threshold t.

too.

Our method also uses some basic knowledge on human face element distribution.

- Eyes are located in the upper part of the face.
- Nose and mouth are aligned on the face vertical symmetry axis.
- Depending on the face orientation, two or only one eye are visible.

- An element cannot be entirely included in another face element.

Figure 2.34 shows a scheme of some basic knowledge used in the anatomic elements extraction method.

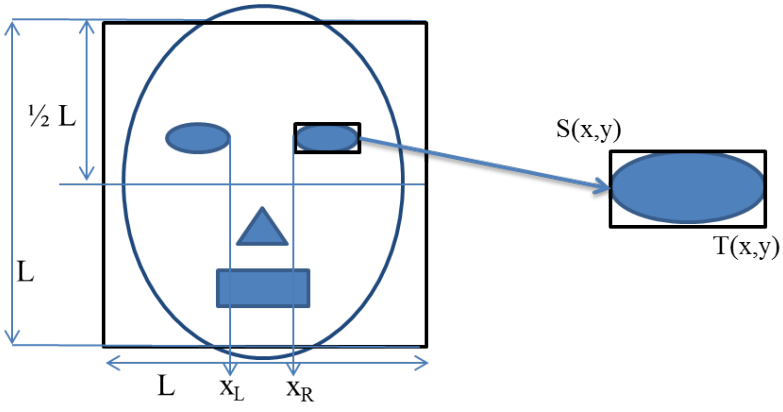

Figure 2.34: **Left:** Scheme of some basic knowledge used in our method. **Right:** An anatomic region bounding box can be represent by the upper left point S and the lower right point T.

In order to show how candidate anatomic regions are extracted, let us fix the threshold at 0.75. As we can see in Figure 2.34, a projection on y-axis of connected components bounding boxes should be able to separate eyes, nose and mouth. So, first we compute the histogram of connected component CC widths according to their ordinate. For each line perpendicular to the y-axis of the face window, the histogram gives the sum of widths of connected component bounding boxes which intersect this given perpendicular line. All connected components with a non empty intersection of their projection on y-axis are merged to form candidate anatomic regions. The Figure 2.35 shows this histogram of widths of connected component bounding boxes, on y-axis.

As we can see in the Figure 2.35, histograms differ according to the face image. It shows that with a fixed threshold, some images may have large connected component such as in image 'd' whereas others have smaller connected components such as image 'b' or 'c'. Components with non empty intersection of their projection on y-axis are merged:

- Image 'a': There are six candidate regions detected from this projection.
- Image 'b': Five candidate regions are detected.

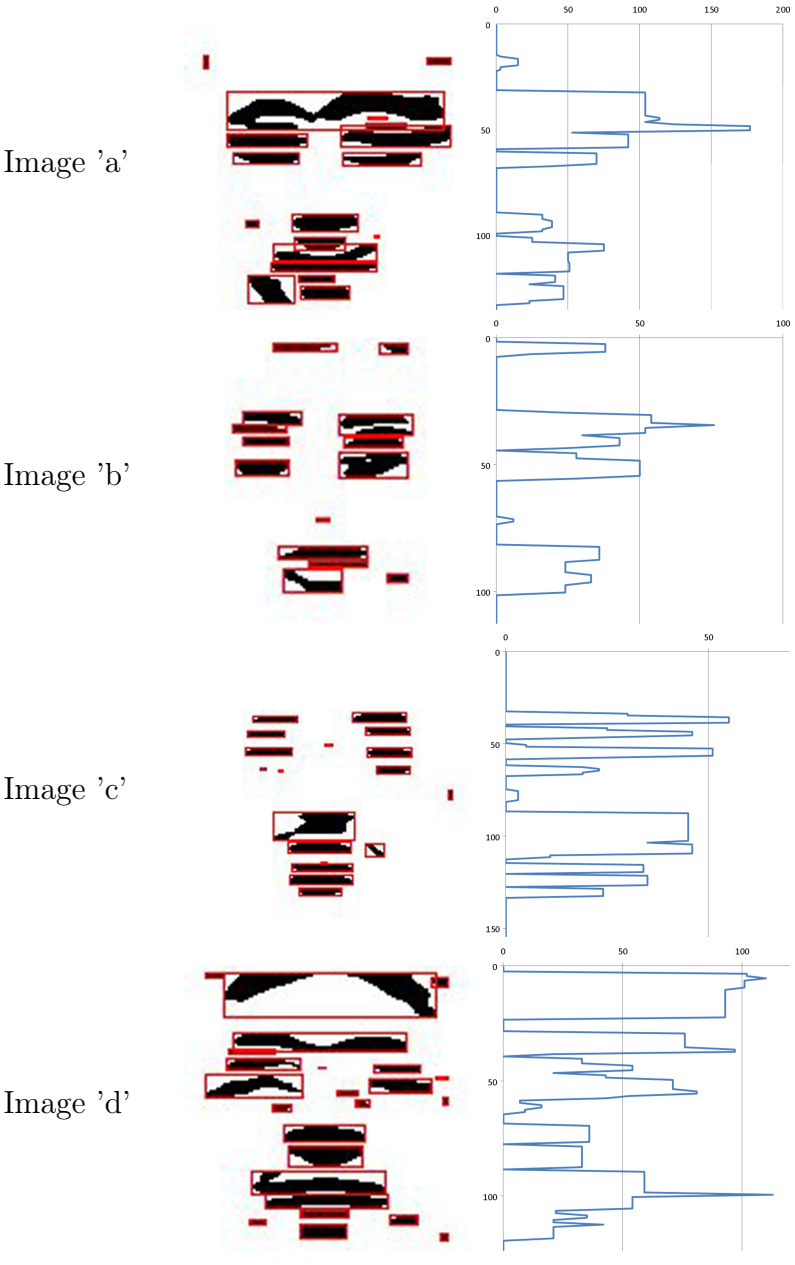

Figure 2.35: Histogram of widths of connected component bounding boxes on y-axis, respectively for face window 'a', 'b', 'c' and 'd'.

- Image 'c': Nine candidate regions are detected.
- Image 'd': Six candidate regions are detected.

Separately, these candidate regions are not significant yet. However, we can notice that a combination of some of them may be representative.

In particular, it seems that the candidate region which includes both eyes is separated from those of nose and mouth.

2.4.4.3 Left and right eye candidate region extraction

At this point, we have generated several candidate regions from projection on y-axis. Here, we want to extract the candidate region which includes both eyes. However, separately, we saw that candidate regions extracted from projection on y-axis are not significant yet. We need to merge some candidate regions to generate the one with the smallest area that will include both eyes. A candidate region is defined by:

- all connected components entirely included in it,
- the upper left point S of this candidate region bounding box,
- the lower right point T of this candidate region bounding box.

First, we assume that two consecutive candidate anatomic regions CAR_i and CAR_{i+1} according to y-axis can be merged if the minimum distance between their projection on y-axis is less than 2.5% of the face window height H. So the number of candidate regions is reduced. For example, after this step, the image 'a' has only 3 candidate regions. Figure 2.36 shows another example.

A basic knowledge of face spatial distribution says that eyes are in the upper part of the face. Since the aim here is to extract eyes candidate regions, only candidate regions the left upper point S of which is located in the upper part of the face are kept. More accurately, the candidate region which is immediately over the horizontal middle line of the face window should contain both eyes. Given y_S, the ordinate of the left upper point of a candidate region, the candidate anatomic region which contain both eyes is the one of which ordinate ${y_S}^*$ is given by the equation 2.26.

$$\begin{cases} {y_S}^* & < & 1/2 \cdot H \\ {y_S}^* & = & \min\limits_{y_S} \left(|y_S - 1/2 \cdot H| \right) \end{cases} \tag{2.26}$$

As a result a candidate anatomic region containing both eyes is selected.

Notice that we extract both eyes in a single region. Then, we have to separate the left and right eyes. As a reminder, the left eye is in the left part of the image (Actually, the eye in the left part of the image is the

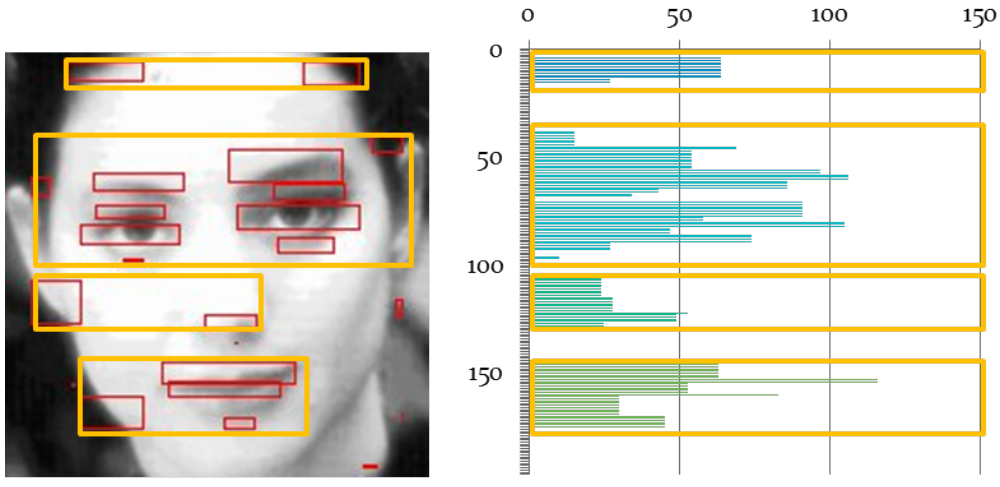

Figure 2.36: Candidate anatomic regions obtained after merging close candidate regions. four candidate anatomic regions are extracted from this histogram.

right eye in the point of view of the concerning person) and the right one is in the right part of the face window.

In order to separate the left and right eyes, the histogram of connected component bounding boxes occurrences is projected on x-axis. Then, two cases are possible. First, eyes projections on x-axis are clearly separated. Second, they are not separated. It happens when face window have a great contrast because of illumination condition, or when the person has glasses or when the threshold is too high. Indeed, when the threshold is too high, the binarization of normalized horizontal energy map shows a single connected component which connects both left and right eye regions because of eyebrow arch. In the example 'a' of Figure 2.35, we can see that both left and right eye regions are connected by the eyebrow arch, whereas in the example 'b', left and right eyes regions are entirely separated. Figure 2.37 shows the associated histograms of occurrences of connected component bounding boxes on x-axis for face windows 'a' and 'b'.

Using the projection on x-axis, new possible regions for a single eye are generated. Connected components with a common projection are merged to form a possible single eye region. The coordinates of left upper point S and of the right lower point T for each possible eyes regions are updated to fit the projection on x-axis. We assume that eyes are the most salient regions in both eyes candidate region. Moreover, there are at most two eyes.

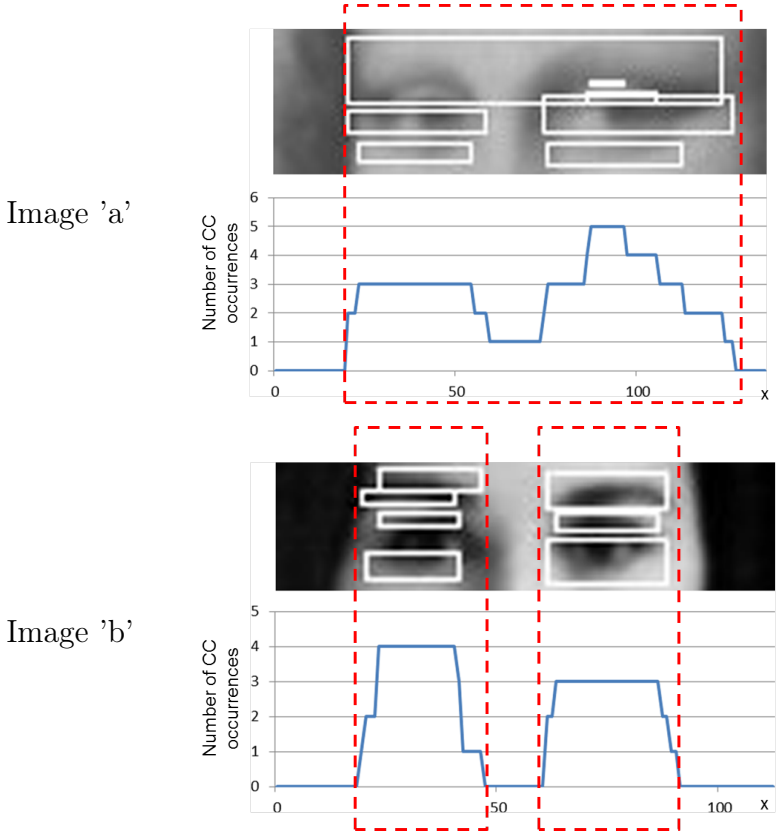

Figure 2.37: Histogram of occurrences of connected component bounding boxes on x-axis.

- Only the possible regions the bounding box area of which is greater than 20% of the possible eye region with the greatest bounding box area are taken into account.
- If there are more than two possible eye regions, only the two possible single eye regions with the largest areas are kept.
- If there is only one possible single eye region, a further step is needed to decide if this single eye region contains a single eye or two.

After this step, if there are still two regions selected, it means both regions are the most salient ones with comparable areas. The possible eye region at the left side of face window will be the left eye candidate region, and the other one will be the right eye candidate region.

However, only one eye region can have been selected. In such cases, this region can contain one single eye or both as we can see in the face window 'a', but only a thin part connects the left and right eyes. So, in order to see

if this single possible eye region contains one or two eyes, we compute the histogram of the connected component on x-axis in this single eye region. In other words, for each abscissa of this single region, the histogram will give the number of pixels of the normalized energy map, the energy of which is smaller than the threshold according to this abscissa. The center part of Figure 2.38 shows this histogram. As you can see, compared to the histogram of connected components occurrences of image 'a' on Figure 2.37, this one is more accurate and gives a better representation of the energy projection on x-axis. However, it is more time consuming than the histogram of occurrences of connected component bounding boxes.

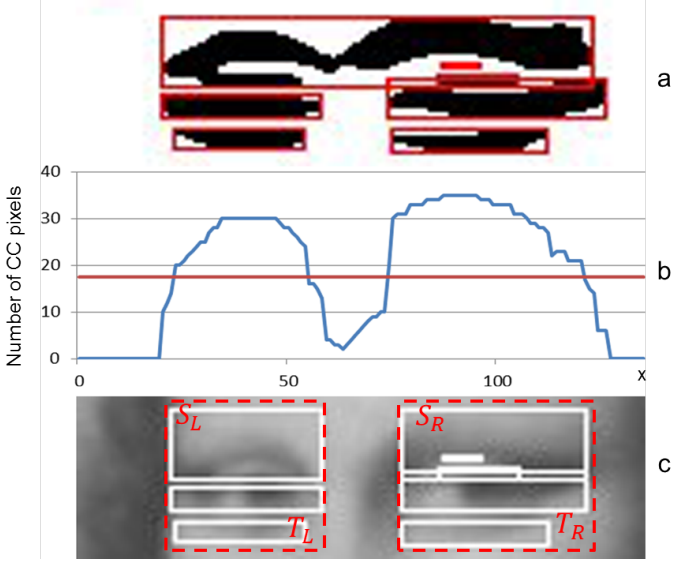

Figure 2.38: a) Initial connected components and their bounding boxes of candidate anatomic region containing both eyes. b) Number of connected component pixels according to the abscissa as well as the upper level set. c) Separation of left and right candidate anatomic regions using this level set.

An upper level set is then applied on this histogram; the level set is associated with half the maximum value of this histogram. Since two eyes will be separated by only a thin part of connected components, such a level set should separate the original both eyes region into two parts. As we can see in Figure 2.38, the support of the level set function is separated in two regions. Similarly to the case when both eyes were found, two single eye regions are generated. Connected components as well as each upper left point S and each lower right point T are then updated to fit each found region. A validation is finally made on each bounding box area to see if one of them is negligible compared to the others. Finally, when these

eye regions are validated, the left region will give the left eye candidate anatomic region, and the right region will give the right eye candidate anatomic region.

2.4.4.4 Nose tip and mouth extraction

Let us call L, the left eye candidate anatomic region and R, the right eye candidate anatomic region.

Once eyes are located, we introduce a new and common knowledge: nose and mouth are located on face symmetry axis: Face has a vertical symmetry axis that passes between the eyes, through the center of the nose tip and the center of the mouth. If the mouth and the nose tip are visible, they should be on that axis. Since we found the candidate anatomic regions of left eye and right eye, we should be able to select connected components that are located on this face symmetry axis. Given x_L the abscissa of the right lower point T_L of the left eye candidate region and x_R the abscissa of the left upper point S_R of the right eye candidate region, both eyes are then separated on x-axis by the interval $K = [x_L, x_R]$. Given CC_i a connected component which does not belong neither to L nor to R, we define NM the anatomic region which gather the nose and the mouth. To define NM, we need to know which CC_i is included in NM. A connected component which belongs neither to the left eye nor to the right eye belongs to NM if it intersects the face vertical symmetry axis. The equation 2.27 tells us whether CC_i belongs to NM or not.

$$\begin{aligned} K &= [x_L, x_R] \\ CC_i &\in NM \Leftrightarrow P_x(CC_i) \cap K \neq \emptyset \\ CC_i &\notin NM \Leftrightarrow P_x(CC_i) \cap K = \emptyset \end{aligned} \tag{2.27}$$

At this point, we should be able to separate the connected components which belong to either the left or right eye or the nose or the mouth from the other connected components. Figure 2.39 shows all the connected component bounding boxes which belong to one of the searched anatomic regions.

Figure 2.39 shows at this point, some connected components located on the boundary of the face are removed. Other connected components located on the eyebrow arch are divided to generated two separated new connected components. Finally, selected connected components are those which are located on an eye or on the nose tip or on the mouth. Notice that

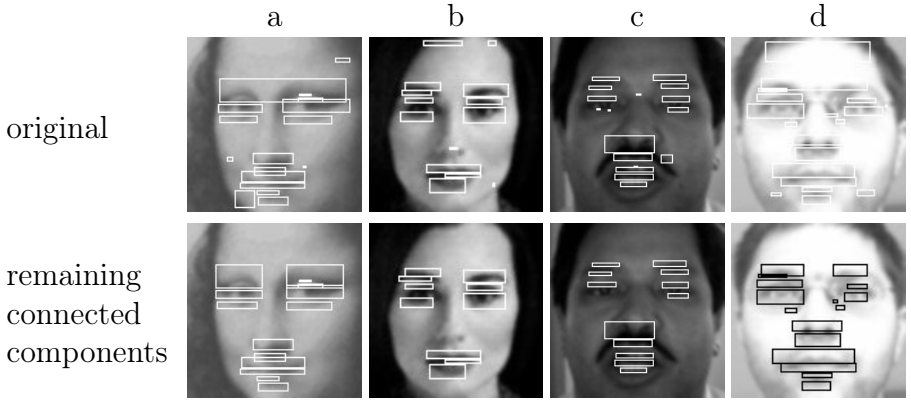

Figure 2.39: Connected components taken into account for a fixed threshold: they should belong either to eyes or to nose tip or to mouth.

for image 'b', no connected component is located on nose although there was a little connected component detected. This connected component was removed in the very first step, when small connected components were removed. It shows that with this fixed threshold, our approach fails in localizing the nose. However, do not forget we only generate candidates, it means, that with a proper threshold, a nose tip region might be selected.

Now that we have selected all connected components representing the nose and the mouth in the region NM, we must separate them into two separate candidate anatomic regions, the first one representing the candidate anatomic region of nose and the second one gathering together components of mouth candidate anatomic region.

Here, we introduce a common basic knowledge: the nose is over the mouth. Since the origin of the image is the left upper corner, it means that the connected component CC_{nose} with the lowest ordinate Y_{nose} belongs at least to the nose. Let us call y_{CC} the ordinate of the left upper corner of the bounding box of a given connected component CC_i. We define CC_{nose} and its associated bounding box left upper corner ordinate Y_{nose} by the formula 2.28.

$$CC_{nose} = CC_i \text{ with } Y_{nose} = \min_{CC_i \in NM} y_{CC} \tag{2.28}$$

Moreover, mouth should be larger than nose, it means that the widest connected component CC_{mouth} should be at least a part of the mouth. Let us call W_{mouth} the width of CC_{mouth} and w_{CC} the width of a given CC_i. Figure 2.40 shows some examples of chosen CC_{nose} and CC_{mouth}.

$$CC_{mouth} = \underset{CC_i \in NM - \{CC_{nose}\}}{argmax} \; w_{CC_i} \tag{2.29}$$

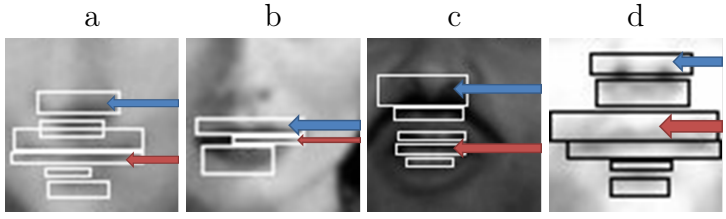

Figure 2.40: Examples of noticeable connected components: the highest one, CC_{nose} is indicated by a blue arrow and the widest one, CC_{mouth} is indicated by a red arrow.

Since the mouth is under the nose, all connected components under the connected component CC_{mouth} should be parts of the mouth. Given M the candidate anatomic region of the mouth and Y_{mouth} the ordinate of CC_{mouth} upper left corner, the formula 2.30 shows how to process all connected components under CC_{mouth}.

$$\begin{aligned} &\forall CC_i \in NM - \{CC_{nose}, CC_{mouth}\}, \\ &\quad \text{if } y_{CC} \geq Y_{mouth} \text{ then } CC_i \in M \end{aligned} \tag{2.30}$$

The last connected components which are located between CC_{nose} and CC_{mouth}, they are classified according to their width. If such a connected component width is closer to CC_{nose} width compare to CC_{mouth}, it will belong to nose otherwise it will belong to mouth.

Finally, Figure 2.41 shows some candidate anatomic regions corresponding to left and right eyes, nose and mouth for a fixed binarization threshold of 0.75 of the normalized horizontal energy map.

As shown in Figure 2.41, all regions are not detected well. In image b, nose detection is wrong; we can observe that nose is located at the upper lip of the mouth. We saw that there was only one single connected component corresponding to nose which was not kept because its size was too small. In image c and d, some eyes are only partially extracted. However, if detection is incomplete, it does not mean that the extraction step failed. Since extraction of candidate anatomic regions is done with a fixed binarisation threshold, a better threshold which is able to extract complete regions may exist. Although some regions are found correctly with a fixed threshold (image a), it seems obvious that it is not the case for many face images.

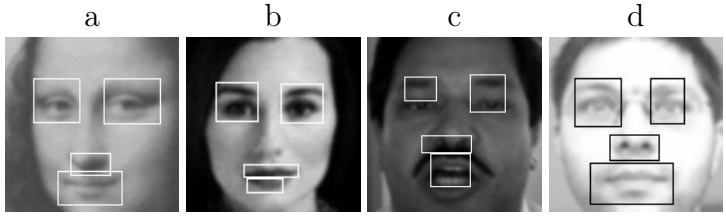

Figure 2.41: Extracted candidate anatomic regions of both eyes, nose and mouth for a fixed binarisation threshold of 0.75 of the normalized horizontal energy map.

2.4.5 Multi-threshold analysis of the normalized horizontal energy map

At this moment, we have shown how to extract candidate anatomic regions; when a specific threshold is applied on the normalized horizontal energy map. Four candidate anatomic regions are extracted corresponding to:

- the left eye candidate region, LE,
- the right eye candidate region, RE,
- the nose tip candidate region, N,
- the mouth candidate region, M.

Figure 2.42 shows some candidate anatomic regions according to the threshold t applied on the normalized horizontal energy map.

As we can see on Figure 2.42, the higher the threshold is, the larger the candidate anatomic regions are. When $t = 0.59$, we can see that candidate anatomic regions in image 'a' and 'd' are correct, whereas this extraction is wrong or incomplete for image 'b' and 'c'. However, with higher values of the threshold, extraction seems more correct, but when thresholds are too high, candidate regions become too large and are not accurate enough. In order to select a suitable candidate anatomic region, we have to study how candidate regions of a given face element vary according to the threshold t and to determine the anatomic region with respect to this variation.

As the threshold varies, the position and the size of candidate regions of a given face element varies too. However, this variation can be more or less important. We assume that a candidate region of a given face element is potentially suitable if it is stable despite the variation of threshold. Instability of candidate region positions and size tells us where connected

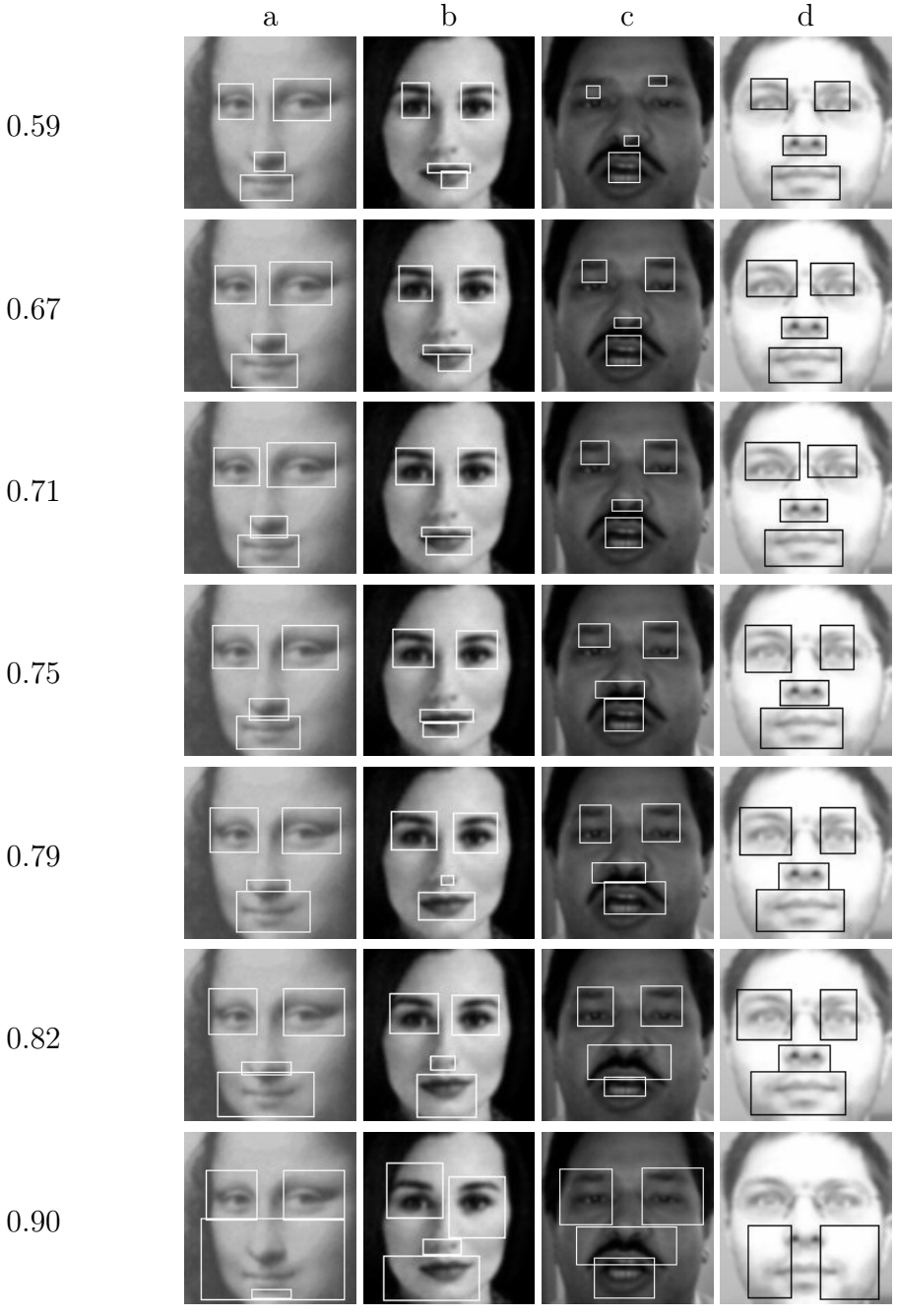

Figure 2.42: Candidate anatomic regions extraction according to threshold t applied on normalized horizontal energy map.

components or regions of different face elements are merging. Indeed, with low values of the threshold, connected components of the same face element will merge as the threshold increases. However with high values of

the threshold, some regions can entirely merge with other regions. This instability with low thresholds values can be ignored since it is the result of connected components of the same face element which are growing or merging. However, when threshold values are high enough, a great variation of position or size implies that different regions are partially or entirely merging with others. As a result, regions are often absurd in terms of positions or size. For example, in the image 'a', with $t = 0.9$, nose is not only too big, it is also represented by a bounding box which entirely includes the mouth bounding box. In the image 'd', eyes are located in the lower part of the face image: this is not possible.

So, we propose a multi-threshold analysis based on these previous remarks. For a given face element:

- First, we want to find where the candidate regions positions and size are stable despite the threshold variation.
- Second, candidate regions must respect some limit values related to their position and size. An eye candidate region or a nose tip bounding box area must be less that 10% of the face window area. The mouth candidate region area must be less than 15% of the area of the whole face image.

Notice that these limitations are very large and are only used to exclude some absurd candidate regions. Do not forget that candidate anatomic regions must also respect some basic knowledge. Eyes must be in the upper part of face window, and two anatomic parts must be separated. So, a candidate anatomic region represented by its bounding box can not include entirely another candidate anatomic region. For example, a nose region cannot be included in a mouth region.

Given an anatomic region R, this region is defined by a set of candidate anatomic regions depending on the thresholds. Each candidate regions is defined by 4 values:

- the abscissa x_R of the left upper point of the bounding box of this candidate anatomic region, normalized by the size L of the face window.
- the ordinate y_R of the left upper point of the bounding box of this candidate anatomic region, normalized by L.
- the width w_R of the bounding box of this candidate anatomic region, normalized by L.

- the height h_R of the bounding box of this candidate anatomic region, normalized by L.

In other words, for a given region R, four functions, $x_R(t)$, $y_R(t)$, $w_R(t)$ and $h_R(t)$, define the evolution of candidate anatomic regions related to R according to the threshold t. Figure 2.43 shows the four functions related to the left eye regions of image 'd'. The abscissa and width were normalized by the face window width W and the ordinate and the height were normalized by the face window height H.

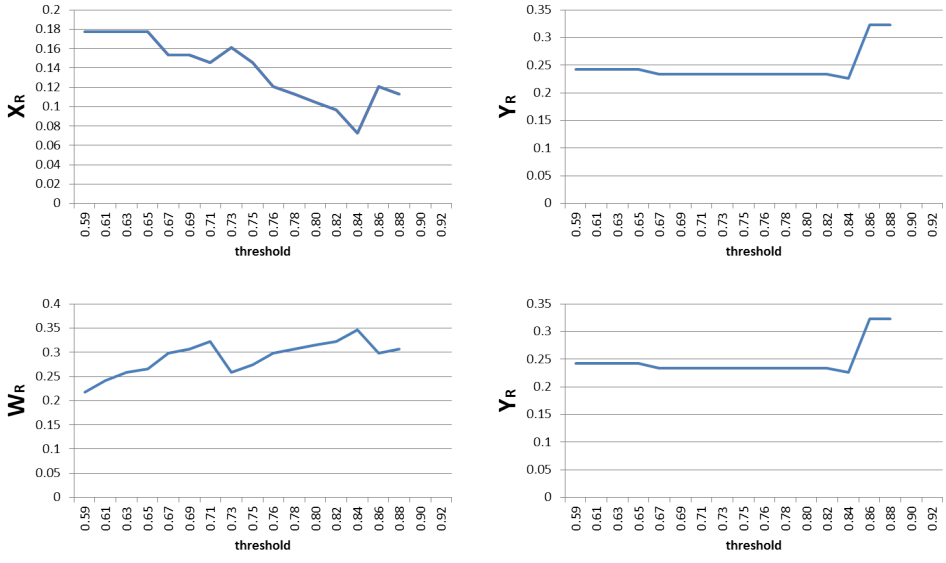

Figure 2.43: Variation of position and size of the bounding boxes of left eye candidate anatomic region for the face of image 'd'.

When we observe the four functions, there are no values for $t > 0.88$. Indeed, when we observe the image 'd' of Figure 2.42 with $t = 0.9$, the eyes are located in the lower part of the face window. Since, eyes must be in the upper part, it is not possible to find the left and right eye for $t > 0.88$. So, for the left eye region of image 'd', only thresholds between 0.59 and 0.88 are taken into account.

In order to select a suitable threshold for a given face element region R, we should also add some criteria on local stability despite threshold variation, as well as maximum area condition.

We define a function D which accumulates the variations of the four functions. The speed of the variation with respect to the threshold value is modeled by the derivative.D is defined by the equation 2.31.

$$D(t) = |x'_R(t)| + |y'_R(t)| + |w'_R(t)| + |h'_R(t)| \tag{2.31}$$

Let us note α the maximum ratio a specific region can have. As we said, $\alpha = 0.1$ for each eye and nose; an eye must have an area less than 10% of the face window area. $\alpha = 0.15$ for mouth; a mouth must have an area less than 15% of the face window area. To express the constraint, we introduce a function A defined by the equation 2.32.

$$A(t) = \alpha \cdot L^2 - w_R(t) \cdot h_R(t) \tag{2.32}$$

$D(t)$ must be low and $A(t)$ must be positive. We need to choose a threshold called ϵ_R and we have chosen ϵ_R is equal to the mean of all $D(t)$.

A suitable threshold t^* related to a specific region R is given by the equation 2.33.

$$t^* = \max_{A(t)>0 \text{ and } D(t)<\epsilon_R} t \tag{2.33}$$

When $D(t)$ is low, it means that candidate anatomic regions vary a little in the neighborhood of the threshold t. However, when $D(t)$ is high, candidate anatomic regions position and size vary a lot. As we said, when $D(t)$ is high, but t is low, it often means that connected components of the same region are merging. This is the reason why we want to maximize t, as long as different regions of the face are not merging.

For a given region R, the suitable threshold t^* respects the stability condition, maximizes the threshold t as long as R does not merge with another region, respects some specific conditions related to positions and size. The left side of Figure 2.44 shows the function D corresponding to the left eye of Image 'd'. The red line is ϵ_R, the mean value of all $D(t)$. The right side shows the left eye candidate region bounding box area variation according to the threshold t. Since w_R and h_R are normalized, the area of the face window is 1. So, the left eye of image 'd' must have an area less than 0.1. the red line in the right graph is α.

In the graph of D, each time $D(t)$ is greater than ϵ_R, we can suppose that it is because connected components of the same region are merging. The chosen threshold will be the maximum threshold which respects the stability and the maximum area condition. In the example of the left eye of image 'd', $t^* = 0.86$.

Similar operations are done for all face regions. Finally, a suitable threshold is computed for the left eye (t^*_{LE}), the right eye (t^*_{RE}), the nose

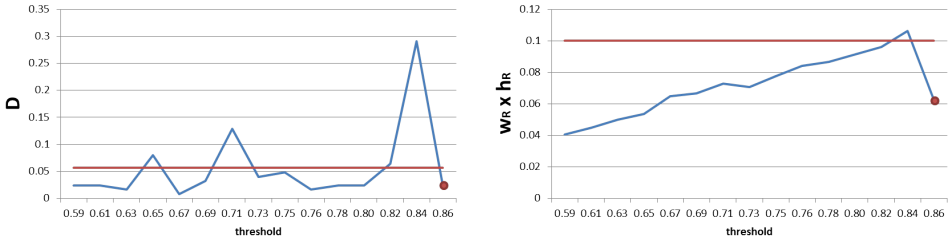

Figure 2.44: Functions D and $w_R \times h_R$ associated with the left eye region area of image 'd' according to the threshold t.

Table 2.3: Selected thresholds respectively for the left eye, right eye, nose tip and mouth regions of images 'a', 'b', 'c' and 'd'.

image	t^*_{LE}	t^*_{RE}	t^*_N	t^*_M
a	0.88	0.80	0.80	0.76
b	0.86	0.86	0.90	0.80
c	0.80	0.82	0.84	0.90
d	0.86	0.86	0.84	0.67

tip (t^*_N) and the mouth (t^*_M). Table 2.3 shows the selected thresholds for each region of the four example images and Figure 2.45 shows the visual results of the selected left and right eye anatomic regions, nose tip anatomic region and the mouth anatomic region.

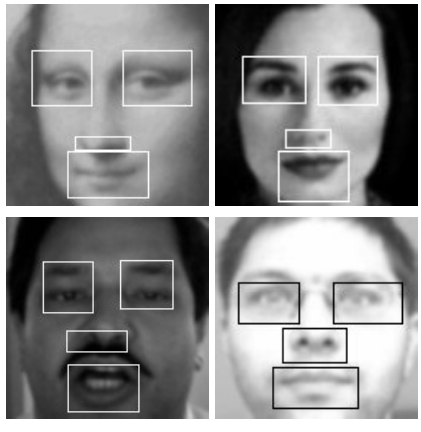

Figure 2.45: Selected candidate anatomic regions: they are the candidate anatomic regions for which we have computed suitable thresholds.

2.5 Evaluation

In this section, our face anatomic regions extraction method will be evaluated. First, we will begin this introduction by presenting the face databases used for evaluation. These databases are mainly, BioID, Color Feret, LFW and CMU/MIT face databases. Then, we will show the results about background extraction using the vertical energy map. Finally, the results about eyes, nose and mouth bounding box extraction will be presented.

2.5.1 Still face image databases

The main face image databases used for this evaluation are BioID and Color Feret databases, because some significant points are labeled and localized. Color Feret database annotates the iris positions as well as the mouth and nose centers whereas BioID database annotates only the iris positions.

BioID is a database containing 1521 images of daily and non controlled environment scenes. Each image contains only one face whereas Color Feret database contains 11338 face images where the lightning conditions and the background are controlled. Both databases have faces from different persons or various ethnic origins. They also contain face images with difficult illumination conditions, with occlusions due to mustaches, beards, glasses and hands. They both contain thousands of images.

LFW is also a widely used face image database. It contains thousands of face images in daily scenes with more complex backgrounds. All faces of LFW databases are detected by the OpenCV implementation of Viola and Jones face detector. Unfortunately, positions of face salient element are not labeled.

Finally, the CMU/MIT face database is also used in this evaluation. This database contains only hundreds of face images of daily scenes at different scales and lightning conditions. Nevertheless, some salient points of eyes, nose and mouth are labeled.

2.5.2 Separating face area from the background

As we said, separating face area from the background should reduce the search area and should delete some noise in the step of face salient element extraction itself. However, in a context of video where there are a lot of images, this preprocessing step must be as fast as possible. Even if real-time is not needed, the important amount of images in a video needs

fast processes. Although an accurate segmentation of the face is desirable, processing time has to be taken into account. Besides, the aim in this part is not exactly the segmentation of the face, but we just want to generate an approximate mask of the background. Accuracy is not the priority here. When our method separates the background from the face area, it should keep a maximum number of face elements. Indeed, if a face element is considered as a background, the main step of face element extraction will be incomplete or wrong. This is the reason why we will classify these masks into three different types.

- The masks in which real face salient elements are partially or completely in the background area will be considered as wrong masks.
- The masks in which segmentation of the face failed, but are not wrong masks will be considered as incomplete masks. There are several cases of incomplete mask. Borders of the face are not well detected; borders are located mostly in the background region of the mask. Sometimes, a border is not detected at all.
- The masks where face borders are mostly well detected are considered as good masks.

Only the wrong masks will be involved in a false result of the face element extraction. Incomplete masks do not involve false extraction. Figure 2.46 shows some examples of good masks. Figure 2.47 shows some examples of incomplete marks and Figure 2.48 shows some examples of wrong masks. In these figures, all the regions in red represent the background.

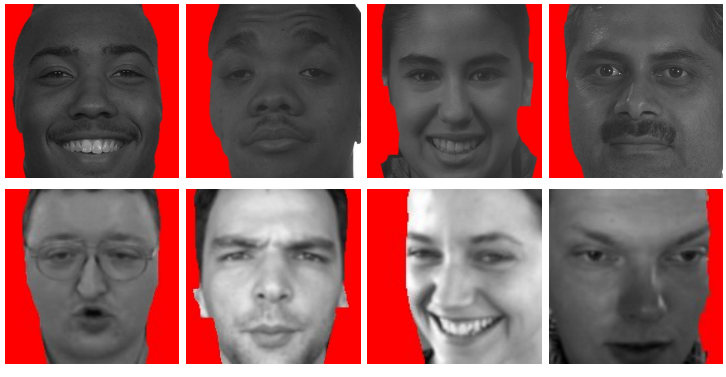

Figure 2.46: Examples of good masks. The first line contains images from Color Feret database and the second line contains images from BioID database.

Figure 2.47: Examples of incomplete masks. The first line contains images from Color Feret database and the second line contains images from BioID database.

Figure 2.48: Examples of wrong masks. All images belong to Color Feret database.

We can see in Figure 2.46 which contains good masks that our method is able to detect only one border if the other one does not exist. As we can see in Figure 2.47, some detected face borders can be totally false, whereas others are only partially false. However, as face salient elements are not considered as the background, it is still possible with these incomplete masks to extract eyes, nose tip and mouth. In wrong masks of Figure 2.48, one or more elements are considered as background. If we were only interested in terms of face segmentation, some of the wrong masks would have been better than the incomplete ones. However, with these wrong masks, some salient face elements are definitely lost for the salient region extraction step. This is the reason why, the notion of quality here is different from the strict segmentation quality notion.

With all these remarks, we can evaluate our method which separates the background and the face. Table 2.4 gives the percentage of good, incomplete and wrong masks using BioID and Color Feret databases. Although

Table 2.4: Evaluation of the mask generation method. The mask should separate the face region from the background.

Database	Good(%)	Incomplete(%)	Wrong(%)
BioID	90.32	9.68	0.00
Color Feret	88.75	10.59	0.66

BioID contains images with uncontrolled background (scene of a person with various background) and Color Feret has a controlled background, we can see that BioID percentage of good masks is higher than Color Feret one. The difficulty does not rely only on the complexity of the background, but also on the complexity of the face. Color Feret has more images with extreme variation of illumination, with beards, mustaches, glasses. Face can be turned, not frontal. BioID faces are almost all frontal. Even if illumination varies from a face to another, in a given face, illumination is more stable. As we said, we want to minimize the number of wrong masks. As we can see in this table, almost all the masks are either good or incomplete. In BioID, there is not such mask. In Color Feret, 99.34 percents of the masks can potentially give a suitable face salient anatomic region extraction.

2.5.3 Anatomic region extraction evaluation

Our problem has not been considered in the same way in the literature, then evaluation is difficult. Usually in the literature, methods try to find only one specific part of the face. This evaluation will essentially concern eye detection, since it is possible to measure detection rate and accuracy of the detection. Unfortunately, it is not the case for mouth and nose detection. The only measurement about nose and mouth detection is the correctness. Our method will be compared to Li et al. method [Li et al., 2008] and to Asteriadis et al.[Asteriadis et al., 2009]. In these papers, the standard of Jesorsky [Jesorsky et al., 2001] is used on BioID to test the accuracy.

Given d_r the distance between the true right iris location and the center of the detected right eye, d_l the distance between the true left iris location and the center of the detected left eye and given d_{rl} the distance between both true iris locations, Jesorsky defines the error err by (2.34).

$$err = \frac{max(d_r, d_l)}{d_{rl}} \tag{2.34}$$

For eye detection task, the reception threshold is 0.25. In other words when $err < 0.25$, the localization is considered to be right. Note that this paper is about detection of eyes and not about their localization. The main difference between detection and localization problems is the first one tries to detect a region whereas the second one tries to locate some salient points. For localization issue, an error threshold of 0.05 or 0.1 is required [Tan et al., 2009]. Jesorsky standard uses iris positions. So first, we have to estimate the iris position.

Our approach of eye regions detection is not designed only for frontal facial views. As a head turns, only one eye becomes visible. Our approach can detect an eye in such cases. However, faces in BioID database are frontal views. So both eyes are visible. We first compute the rate of images where both eyes are not detected and found that it is less than 0.0034%. Since Jesorsky measure needs both eyes, only images where there are both eyes are taken into account. In other words, 99.9966% of BioID database images are considered for the evaluation.

In our approach of eye detection, horizontal lines on faces are detected. Since eyebrows and eyebrows arch are also horizontal, they are systematically detected in both eyes. Since the aim is to detect bounding boxes of face salient anatomic regions, we assume that eyebrow may be relevant and thus, can be incorporated in eye region. However, since Jesorsky standard considered eye center as the iris and since our eye detection includes eyebrow or eyebrows arch, the coordinates of point C at the estimated center of the eye is given by the equation (2.35). x_E, y_E, w_E and h_E are respectively the abscissa, the ordinate of the left upper point of the eye region bounding box, the width and the height of the eye region bounding box.

$$\begin{aligned} x_C &= x_E + \frac{1}{2} \cdot w_E \\ y_C &= y_E - \frac{1}{3} \cdot h_E + \frac{1}{2} \cdot \frac{2}{3} \cdot h_E \end{aligned} \tag{2.35}$$

Table 2.5 compares our method with Li's et al. and Asteriadis et al. methods. Note that our method has a better correctness whereas Li's one has a better mean error. Despite the approximation of eye center position we used, results on BioID database are quite similar.

Our method is based on selecting suitable candidate among candidate anatomic regions of a same salient region. The first question is to measure the interest of the multi-threshold analysis in the detection of the eyes. Therefore, table 2.6 gives the correctness and the relative mean-error according to threshold as well as the correctness and mean error of the

Table 2.5: Comparison of Li et al. and Asteriadis et al. methods with ours on BioID database.

Method	Correctness (%)	Relative mean-error
Li et al.	96	0.1004
Asteriadis et al.	96	Not given
Our method	97.23	0.1130

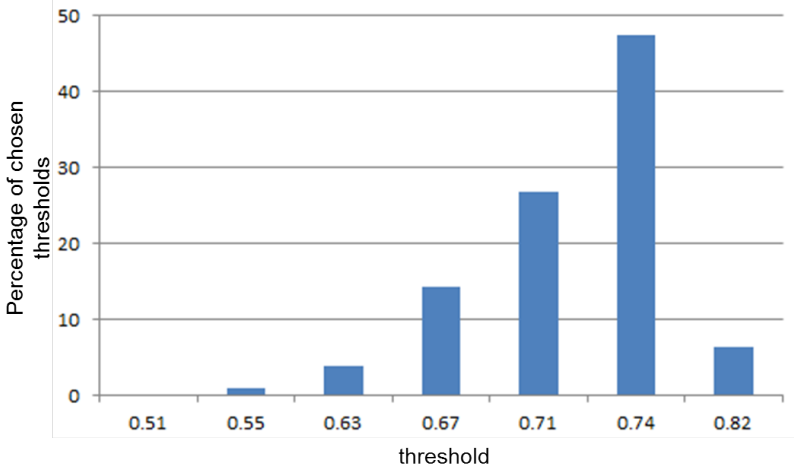

Figure 2.49: Percentage of chosen thresholds by the multi-threshold analysis.

multi-threshold approach.

As shown in table 2.6 the correctness ($err < 0.25$) rate is not equally distributed. One can observe a maximum correctness with low mean error for a threshold value close to 0.74. This is the effect of the normalization of the horizontal energy map. This normalization reduces the number of thresholds we have to study. Note that from a threshold higher than 0.86, correctness and mean error decreases fast. Indeed, if the threshold is too high, connected components of different parts of the face are no longer separated. This phenomenon is brutal, since a very little higher threshold can merge 2 distinct candidate anatomic regions. Without the multi-threshold analysis, for a threshold fixed at 0.74, the eye detection works with a quite good accuracy and correctness. However, clearly, the multi-threshold approach shows better results in terms of correctness and accuracy than a fixed threshold approach.

Figure 2.49 shows which of the 7 thresholds are chosen by the multi-threshold analysis for eye detection. Note that the most chosen threshold

Table 2.6: Correctness and relative mean-error of candidate anatomic regions with a fixed energy map binarization threshold as well as the multi-threshold approach.

Threshold	Correctness(%)	Relative mean-error
0.39	58.98	0.3872
0.43	62.47	0.3495
0.47	67.95	0.3081
0.51	72.90	0.2768
0.55	77.37	0.2540
0.59	82.79	0.2127
0.63	88.48	0.1773
0.67	91.67	0.1549
0.71	93.90	0.1417
0.74	**95.39**	**0.1309**
0.78	93.43	0.1434
0.82	87.80	0.1792
0.86	72.22	0.2786
0.90	45.60	0.4971
multi-threshold	**97.23**	**0.1130**

is the value where the detection has greatest correctness and lowest mean error. However, only 47% are chosen with this threshold. It also shows that multi-threshold analysis favors lower thresholds than higher. As we said, the lower the threshold is, the more separated candidate anatomic regions are.

Our method was also evaluated on Color FERET database. It contains various images under different conditions of pose, illumination. Many people have beard or glasses. Eyes, nose tip and mouth position are given in all frontal images (about 1800 images). Our method gives similar results: a correctness of 97.60 for a mean error of 0.1110.

We also wanted to evaluate our method on LFW database. Unfortunately, anatomic parts are not labeled or some ground truth such as those used in [Tan et al., 2009] are not public.

About nose and mouth, just the correctness was simply tested. Contrary to eyes, our approach is designed to give a nose and mouth candidate anatomic regions, since at least a part of them must be visible. Table 2.7 gives the ratio of detected nose and mouth in Color FERET and MIT/CMU, since nose tip and mouth center are labeled in these databases.

Table 2.7: Nose and mouth detection rate.

Database	Nose(%)	Mouth(%)
Color FERET	75.23	97.50
MIT/CMU	83.63	97.76

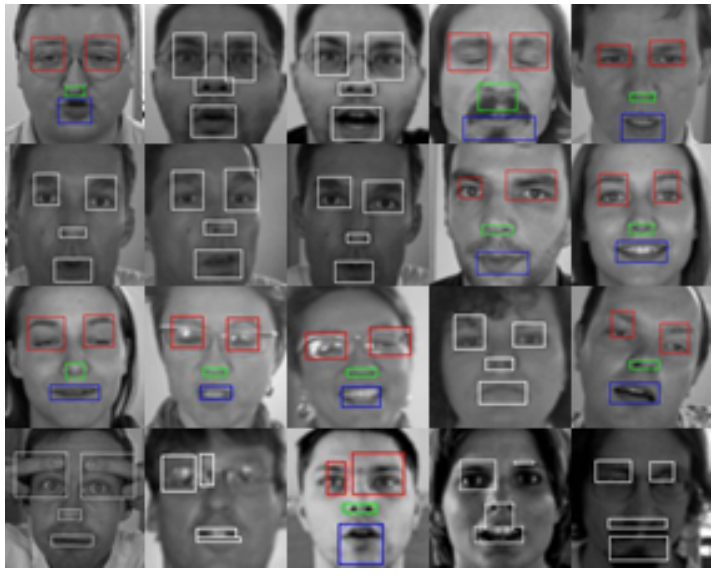

Figure 2.50: Visual results on BioID. Incomplete or wrong extractions are in last line.

For these tests, we assume that a region is correctly found if its bounding box contains the corresponding ground truth point with constraints on its area. Nose region area must be less than 5% of the face window area, and mouth region less than 8%. This table shows that mouth is correctly detected. Nose detection failed when faces have mustache or beard.

The average computation time of our method is 16 ms on Intel Core i7-2670 QM CPU at 2.2GHz. Our software is made for test and thus, is not optimized. Moreover, extraction part is the most important one in the algorithm, but can be done in parallel processes. Otherwise, the computation of each value in the horizontal energy map is quite fast, since Viola and Jones integral image is used. The computation of multi-threshold analysis is the fastest part of our approach. Indeed, we use 7 thresholds. For each threshold, our method detects 4 regions and each regions generates 4 values (position and size). So, only a fixed number of calculations with a data array of 112 values is needed.

Figure 2.51: Visual results on Color FERET. Same disposition as Figure 2.50.

2.6 Conclusion of face salient element extraction

In this paper, we proposed a new method which uses a single adaptive horizontal Haar feature to extract bounding boxes of both eyes, nose and mouth. Knowing the observation level of faces, we are able to detect facial parts with an efficient horizontal energy map. We have also shown how basic knowledge of facial distribution can improve facial anatomic regions detection. Moreover, we propose a multi-threshold method which is able to choose a suited threshold for each part of the face, despite difficult illumination conditions. Evaluation shows the efficiency of a multi-threshold analysis. Our method is also able to detect eyes with accuracy, despite the approximation related to eyebrows or eyebrows arch. Mouth is also well detected whereas nose is still more difficult to extract, especially when faces have mustache or beard.

Chapter 3

Head pose estimation using Haar energy and face salient elements

Chapter contents

3.1 Introduction 84
3.2 Head pose estimation state of the art 86
3.2.1 Methods based on templates 86
3.2.2 Methods using classification 87
3.2.3 Geometric methods 88
3.2.4 Methods using flexible models 89
3.2.5 Nonlinear regression methods 90
3.2.6 Methods based on Embedding 91
3.2.7 Hybrid methods 92
3.2.8 Discussion 92
3.3 The proposed face pose estimation methods . 94
3.3.1 Estimation of the roll 94
3.3.2 Yaw and pitch estimation 103
3.4 Evaluation of our pose estimation method . . 112
3.4.1 Evaluation of the roll estimation method 112
3.4.2 Evaluation of yaw and pitch estimation method . 117
3.4.3 Test database 117
3.4.4 Parameters 117
3.5 Conclusion of pose estimation 121

Chapter summary

An image of a face is a projection of the 3D object in image plan. Therefore, according to the orientation of the face in the 3D world with respect to the camera position, two projections of the same face will vary a lot. These variations involve a deformation of the global aspect of the face. Moreover, other variations related to the environment, such as illumination or occlusion, can change the appearance of the face. Furthermore, face is not a rigid object; it means that non linear transformation can also affect face appearance. Despite all these difficulties, face pose estimation tries to give the orientation of the face, according to the degrees of freedom allowed to face movement.

Many methods or applications involving face analysis in computer vision needs to estimate the pose before achieving their own task. For example, the efficiency of many approaches in driver assistance, human computer interaction, face recognition, face tracking depend on the head pose estimation.

In this chapter, we will present two methods to estimate head pose using horizontal Haar energy as well as face salient elements. The first method estimates the roll and the other one estimates the yaw and the pitch.

Résumé du chapitre

L'image du visage est une projection d'un objet 3D sur le plan de l'image. Par conséquent, en fonction de l'orientation du visage dans le monde 3D par rapport à la position de l'objectif de la caméra, deux projections du même visage varieront beaucoup. Ces variations impliquent une déformation de l'aspect global du visage. De plus, d'autres variations liées à l'environnement, comme l'illumination ou l'occlusion, peuvent changer l'apparence du visage. D'ailleurs, le visage n'est pas un objet rigide ; cela signifie que des transformations non linéaires peuvent affecter son aspect. Malgré toutes ces difficultés, l'estimation de pose cherche à donner l'orientation du visage en fonction des trois degrés de liberté que le mouvement du visage permet.

De nombreuses méthodes et applications liées à l'analyse du visage dans le domaine de la vision par ordinateur requièrent d'estimer la pose avant d'exécuter leur propre tâche. Par exemple, l'efficacité de nombreuses approches telles que l'assistance à la conduite, l'interaction homme-machine,

la reconnaissance du visage ou son suivi dans la vidéo dépend de l'estimation de pose.

Dans ce chapitre, nous présenterons deux méthodes utilisant l'énergie de Haar ainsi que les éléments saillants du visage pour en estimer la pose. La première estimera le roulis tandis que la seconde estimera le lacet et le tangage du visage.

3.1 Introduction

The head pose estimation consists in estimating the orientation of a face according to a model generally adopted and which considers the three possible rotations of the object in the real world. Let us consider a frontal face, these rotations are:

- the roll, which is the rotation of the face in the image plan,
- the yaw which corresponds to the rotation when the head turns to the right or to the left.
- the pitch which corresponds to the rotation when the head moves up or down.

The Figure 3.1 shows the orientation of the roll, the yaw and the pitch on a face.

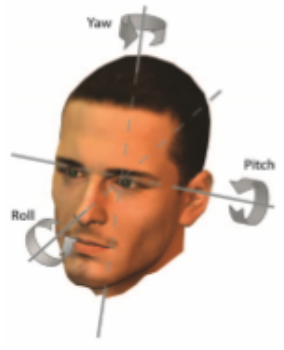

Figure 3.1: Visualization of the roll, yaw and pitch of a head.

The image of a face is a projection on the image plan of the three dimensional face object. As a result, face images of a same person vary a lot. The pose variation makes all computer visions tasks difficult. Although it is not the only difficulty, knowing the pose may be useful to many processes.

Besides, all rotations are not equivalent in terms of pose correction. Indeed, if we want to generate a frontal view of a face (with null roll, yaw and pitch values) from another view of the same face, but with a different pose, the roll can be corrected by a simple rotation in the image plan, contrary to the yaw and the pitch corrections which require complex transformations.

New human-machine interactions are designed, based on human movement. In such cases, finding the orientation of objects is important to

properly localize the objects in three dimensions, for example, in the field of video games with the well-known Kinect, or in driver's fatigue control systems. In all these interactions, pose has to be estimated.

In this chapter, we have chosen to begin by presenting a method which is able to estimate the roll of a face, because it can be corrected with a single transform. This method will use the horizontal global energy of a face, a scoring system extended from the salient anatomic elements extraction method developed in the previous chapter. Then, a geometrical method based on the extracted anatomic regions, which computes the yaw and the pitch will be presented. In our approach, the computation of the roll, the yaw and the pitch are independent. All these methods are geometric. However, roll estimation is based on the presence of horizontal elements in a face, whereas yaw and pitch estimation are based on spatial face elements distribution.

The goal in this thesis is to select a "good" face candidate image among those present in a video sequence. We assume that a good face candidate should have null roll, yaw and pitch values. Therefore, an accurate estimation for all poses is not required. The pose estimation should be accurate with only almost frontal faces. Moreover, the pose estimation method should be computationally fast because any video sequence will generate a large amount of images.

After we present a brief state of the art where so many different methods have been developed, we will introduce two original methods we defined to measure, on the one hand, the roll and on the other hand, the yaw and the pitch of human face. The last part of this chapter will end with the evaluation of these methods.

3.2 Head pose estimation state of the art

This section will present some of the main works in head pose estimation. Approaches to estimate head pose depend on the type of algorithms used. As we said, while some methods only estimate one or two degrees of freedom, others will estimate all of them. However, whatever are the degrees of freedom estimated, they all respond to the same problem of face pose estimation. In this state of the art, pose estimation methods were divided into seven categories.

- Methods based on templates: A test face image is compared to a set of face images of a database where each face is labeled with a discrete pose. These methods assume that similar images will have similar pose.
- Methods using classification: Several sets where each of them is labeled and contains a pose are used to train a machine learning technique.
- Geometric methods: They use some basic knowledge of face elements distribution and location to determine the face pose.
- Methods using flexible models: These methods try to learn the continuous deformation of faces. Features or some parameters of the deformation model are used to estimate the pose
- Nonlinear regression methods: They use linear and nonlinear functions in order to map a set of features to a head pose.
- Methods based on embedding: They assume that head pose continuous variation can be modeled by a low-dimensional manifold. Then, these manifolds are used for regression.
- Hybrid methods: They combine some previous methods. They assume that a combination can overcome some limitation proper to each method.

3.2.1 Methods based on templates

Some of the first pose estimation methods are template methods. Given a face test image, they try to estimate the pose by comparing this query image with samples of the set where pose is labeled for each sample. These methods compare the whole face with others by using a metric. Then, the

pose is determined using the most similar templates pose. In [Beymer, 1994], the authors propose to use normalized cross-correlation at multiple image resolutions. In [Niyogi et Freeman, 1996], the authors use mean square error over a sliding window to compare features. The main advantages of these methods are, first, the template set can be incremental: a new face template can be added at any time. Second, such methods do not require negative examples.

However, there are also important drawbacks. First, with these methods, pose estimation is too sensitive to face alignment. Furthermore, as the template set grows, computation time to estimate the pose of a query image will increase too much. These methods assume that the similarity between two persons of the same pose should be less than other comparisons of two other face images. This assumption is not always exact. If we consider two face images of the same person at different poses, the similarity between these images can obviously be higher than the similarity between two images of different persons but with the same pose.

3.2.2 Methods using classification

Many methods assume frontal face detection. Naturally, if they succeed in finding faces in a frontal view, they could extract faces in a defined discrete pose. Methods using classification will train multiple face detectors. Each of them is associated with a specific pose. Some of these methods assume a face is detected by one of these detectors, it means that the associated pose will be the one of the test face image. However, these methods also assume that two detectors are not able to find the given test face image. Others assume that the selected face detector and its associated head pose will be the one with the greatest support. The main differences between these methods will be on the descriptors and the machine learning techniques.

In [Jones et Viola, 2003], Haar-like features and AdaBoost are used to get several face detectors depending on the pose. One of the first face pose estimation method uses three SVM in [Huang et al., 1999], and tries to classify face images into three different discrete yaws. In [Zhang et al., 2007b], naive Bayesian classifier and a Hidden Markov Model are used to aggregate face pose estimation from several cameras.

The main advantage of such methods is that they are able to detect, localize and estimate the pose at the same time. Indeed, most of the classification methods used face detectors according to a pose. However, such methods have an important drawback. They require training many detectors, each of them corresponding to a discrete pose. Therefore, they

need many datasets which gather together faces with a discrete pose and non face examples, as well as negative examples of faces (i.e. with another discrete pose). The amount of image to annotate and align is huge. On the other hand, these methods are limited by the number of head poses. Indeed, when the number of poses increases, the difference between two consecutive poses will decrease. As a consequence, a training set will have some positive examples of face with a given pose as well as some negative examples of face with a slightly different pose. The appearance between these images will not be enough significant. This is the reason why, these methods often estimate the pose on one degree of freedom, using less than twelve detectors.

3.2.3 Geometric methods

These approaches are based on human perception of head pose. Human being perception is not able to estimate the pose with exact orientation values. However they are able to compare the pose of two heads. For example, in [Wilson et al., 2000], the authors show how human perception of head pose is based on nose deviation as well as the deviation of the face symmetry axis. In general, face elements projection spatial distribution suggests to the human vision what the head pose is. These methods exploit the common knowledge on human face when pose varies. Geometric approaches use these properties of human face to estimate the head pose.

In [Gee et Cipolla, 1994], five points are extracted (each eye outer corners, mouth outer corners and nose tip) the face symmetry axis is considered as the line between the midpoint of the eyes corners and the midpoint of the mouth corners. The nose tip position is then compared to this axis to estimate the yaw of the head. In [Wang et Sung, 2007], the authors extract three lines. The first one links the outer eye corners. The second one connects the inner eye corners and the last one connects the outer mouth corners. They assume that these lines are parallel and thus, if lines are no longer parallel, it is the result of perspective distortion. The vanishing point of these lines can be computed and used to compute the head pose. Geometric methods are fast and require only a few features. However, these features are not easy to be extracted accurately. Moreover some of these features can be invisible because of glasses or any other occlusion.

In [Kong et Mbouna, 2015], a 3D face of a person is used as a model. Then, the authors try to minimize the disparity of some facial features between the query image and the 3D model. In [Dahmane et al., 2010], the authors show that the study of local symmetry of some face regions, as well as the location of face vertical symmetry axis can improve pose

estimation.

3.2.4 Methods using flexible models

Contrary to other methods which try to detect a face with its pose in a rectangular window, methods using flexible models try to create a non-rigid model which describes face structure. In addition to the pose label for each face image, these methods also need the annotation of further structural information. In other words, some additional features have to be annotated in images of training data. Such methods enable to compare features and not the global face window. Most of them use control points (eye corners, nose corners, mouth corners, some face contour points, etc). These points are represented in a deformable graph. These methods assume that points can converge to a graph corresponding to every face by deforming the general model. From the training dataset, most of these methods will first extract these control points and then try to estimate the pose. In [Lanitis et al., 1995], the authors present the Active Shape Model (ASM) in the context of face applications. A principal components analysis on all control points positions of the training data is applied to learn the possible deformations of faces. In [Jiang et al., 2012], ASM is used to first extract some control points, then, using the positions, a SVM is used to estimate the pose. In the ASM, only the positions were considered in the training step, the Active Appearance Models (AAM) consider the position and the texture for all control points. In [Cootes et al., 2000], the authors use the AAM to extract a few control points and assume that the model parameters are related to the pose and hence estimate the pose of a test image using the extracted control points. In [Martins et Batista, 2008] or in [Dai et Chung, 2011], the authors also extract control points using AAM respectively to estimate the pose in a context of a single view image and in the context of video frames.

AAM can localize a head with a small error. However, they cannot be used as a face detector. AAM will never verify the existence of a face in the image. They are only able to converge to the face. In other words, even if a face does not exit in the image, if we place the initial control points on this image, the model will converge by minimizing a distance value and thus find a face. Similar to all techniques with a machine learning step, AAM need to annotate all control points in all faces in the database. Moreover, AAM needs to place all control points, even if they are no longer visible. For example, in almost profile view of faces, one of the eye will no longer be visible, AAM will then fail in localizing all control points.

Many researchers develop similar methods to localize control points on

faces such as in [Luu et al., 2011] where the authors use the Modified Active Shape Model (MASM) described in [Seshadri et Savvides, 2009]. In this paper, the authors propose to combine landmarks for the MASM and some face texture information in the proposed Contourlet Appearance Model to estimate the age. In **??**, the authors propose a statistical model called Statistical Facial Model (SFAM) able to learn the global 3D face deformation as well as the variation of texture and shape around some 3D face landmarks. Such 3D face lankmarking techniques are generally more robust to pose and lighging variations.

3.2.5 Nonlinear regression methods

Nonlinear regression methods try to estimate pose using a non linear function which matches face images to one or more directions. These approaches assume that a function built from the images of the training step, can estimate the pose of any new face image. The main issue of such methods is how well a regression will learn a suitable matching between face images and the pose. Indeed, the amount of features generated in a face image can be very large.

Some methods use regression tools. In [Murphy-Chutorian et Trivedi, 2010], the pose is estimated using localized gradient orientation histograms on support vector regressors (SVRs). In [Li et al., 2004], eigenface and SVRs are used to detect face as well as the face pose. However, in order to use SVRs, the dimensionality of the features must be reduced, using for example Principal Component Analysis. Other nonlinear regression methods use neural networks. For example, in [Bishop, 1995], in [Yang et al., 2012] or in [Voit et al., 2007], Multi-Layer Perceptron (MLP) is used to estimate pose. MLP updates each node weight backward through each layer and the output nodes correspond to discrete pose. In these cases, MLP can only provide coarse estimation of discrete pose. Another popular neural network is the Locally-Linear Map (LLM) built with a lot of linear maps, such as in [Rae et Ritter, 1998] or in [Krüger et Sommer, 2002]. Many pose estimation methods need a first step of face image alignment. In [Haj et al., 2012], the authors use a partial least squares (PSL) regression to estimate the pose and the alignment. These methods are very fast and give accurate head pose estimation. However, pose estimation is reliable only when head is almost perfectly aligned and localized. In [Osadchy et al., 2007], the authors add to MLP a convolutional network which should reduce this drawback.

3.2.6 Methods based on Embedding

Embedding methods assume that a head sample is represented in a high-dimensional space where only a few dimensions vary as the pose changes. Indeed, only the dimensions for the pose and three others for the position are necessary for a rigid object. Hence, a low-dimensional continuous manifold associated with pose variation may represent each high-dimensional image sample. In order to estimate the pose of a test face image, first, this manifold must be built and then an embedding algorithm will project this test image into this manifold. Finally, either the pose is estimated using regression or by matching the result of the embedding or with classification process. In other words, all algorithms which try to reduce the dimensionality can be seen as a manifold embedding. However, all variations which may appear in a face are not only because of pose variation. Such methods need to reliably consider only variations from pose while ignoring the others.

PCA and the Kernelized Principal Component Analysis are widely used. These techniques reduce the dimensionality of the initial features, while conserving most of the information [Duda et al., 2001]. For example, in [Sherrah et al., 2001], Gabor wavelets are used to build embedded templates, then a PCA is applied on these templates to estimate the pose. The main drawback of PCA or KPCA is that these techniques are an unsupervised reduction of the dimension. Hence, although reduced features contain almost all the initial information, we cannot be sure that this information is related to the pose, rather than other variations which may appear.

To overcome this, others have the idea to separate the training data into different sets where each of them is associated with a defined discrete pose. PCA or KPCA are then applied in each set. A further step is then necessary to select the set as well as the associated discrete pose for estimation. In [Srinivasan et Boyer, 2002], pose specific eigenspaces are computed. Then each test image is normalized and projected into each pose-eigenspace. The estimated pose will be the one which maximizes the projection energy. Pose eigenspaces, like the classification methods learn from sets of discrete pose, so it has almost the same drawback: Fine pose estimation is not possible. In order to have finer results, it is better to use Multi-class Linear Discriminant Analysis (LDA), in particular its kernelized version KLDA, such as in [Wu et Trivedi, 2008] where KLDA tries to find how the data vary between two sets of discrete poses. In [Hoffken et al., 2013], the multiclass Linear Discriminant analysis (M-LDA) is used to estimate the pose, while minimizing the impact of the information which does not

concern the head pose.

Some approaches such as Isometric feature mapping (Isomap) [Raytchev et al., 2004], Locally Linear Embedding (LLE) [Roweis et Saul, 2000] or Laplacian Eigenmaps (LE) [Belkin et Niyogi, 2003] tends also to remove irrelevant dimension related to other variations than pose variation.

Unfortunately, these methods tend to build manifolds for both identity and pose. Others suggest to build a separate manifold for each person of the training set [Yan et al., 2008b], but, in the real world, it is difficult to get images of each person with many discrete poses.

3.2.7 Hybrid methods

These methods try to overcome a drawback of one specific method with another approach of another method category. Many methods associate a geometric approach with another approach based on a flexible model, such as in [Hu et al., 2004]. Others use PCA embedded template with optical flow [Zhu et Fujimura, 2004] or with a density Hidden Markov Model [Huang et Trivedi, 2003]. Hybrid methods can also apply a few pose estimation techniques independently. Then, results are merged to get a better estimation of the pose. For example, in [Wu et Trivedi, 2008], manifold embeddings are used followed by Elastic Graph Matching to improve pose estimation.

3.2.8 Discussion

The first attempts to estimate head pose were methods using the templates. These methods try to classify a test face image by comparing it with other labeled annotated images. They assume that faces of a discrete pose are globally almost the same. This assumption is not entirely exact. Indeed, when we analyze the whole face, face samples of the same person with different poses can have a global appearance more similar than two faces of different persons but with the same pose. Therefore, these approaches are almost abandoned.

Then, classification approaches as we described in section 3.2.2 and exclusive geometric approaches were developed. Classification methods can be seen as a multi-class face detection problem, each class representing a discrete pose. They cannot give accurate estimations and require annotating a large amount of images. Exclusive geometric methods use some geometric properties to estimate the pose. They are fast but are in general not so accurate.

Classification methods as we defined, are now rarely used. Now, the trend is to build some models or to find the most representative features according to face pose. Flexible models can describe the deformation of face appearance, generally using control points according to the pose. These methods are visually impressive but are quite difficult to generalize and thus, they depend on the learning database. Others try to find a subset of some features associated with face pose variation. They use embedding techniques or regression methods.

Now, the methods are rarely exclusively geometric, they are actually hybrid methods: face geometry is used as cues for other descriptors to improve head pose estimation.

In the next section, we will present two methods to estimate the pose: the first one estimates the roll whereas the other gives an estimation of the raw and the yaw. Both of the methods are geometric. However, roll estimation method is also statistical whereas the other one is an exclusive geometric approach.

3.3 The proposed face pose estimation methods

In this section, we will present our face pose estimation method. Actually, this method is divided into two independent parts. Some methods estimating yaw or pitch require a null roll face image. Therefore, face roll estimation must be the first of all tasks with methods related to face analysis and which are not invariant in rotation. The first one is mostly a statistical geometric method which is able to estimate the roll of a given face. The second method is geometrical and will estimate the yaw and the pitch. We still suppose that a face is found in a square window L. This window contains the entire face. The order of magnitude of the face is given by the size of the window.

3.3.1 Estimation of the roll

Many of the methods on face detection and recognition, face tracking, with face control points assume a null roll value. For example Viola and Jones face detector needs a face roll value less than $\pm 15\,\text{deg}$. It is almost the same for our face salient element extraction method. After we motivate the type of the approach with propose and analyze the properties of the horizontal energy map with respect with the roll measurement, we sum up the approach [Pyun et Vincent, 2015] and then come to the details.

3.3.1.1 Motivation

One of the main assumption of the face salient element extraction method is that face roll is almost a null value. Although this assumption can be seen as a limitation of our method, it also means that when the extraction succeeds, we can be quite sure that the face has almost a null roll value. Hence, given a test face window I, given $Rot(\alpha)$, the rotation operator of angle α and the center of which is the center of I, depending on α, the extraction will give more or less credible regions. The credibility of regions can be partially evaluated because we know what a face must be. However, this side effect of the extraction method cannot be enough to estimate the roll for two main reasons.

1. When two face images are differentiated by a small rotation, they will probably give almost the same salient elements. Therefore, with small roll variation, it will be difficult to judge which one is the best.

Figure 3.2 shows results of face elements extraction according to the roll value α. As we can see, extraction is credible for roll values between -15 deg to 15 deg. Hence, the roll cannot be estimated accurately if we only consider the credibility of the extraction.

2. On the other hand, for each discrete pose to be tested, salient elements must be extracted. Even if the extraction is fast, as we add discrete roll test values, computation time will also be proportionately longer.

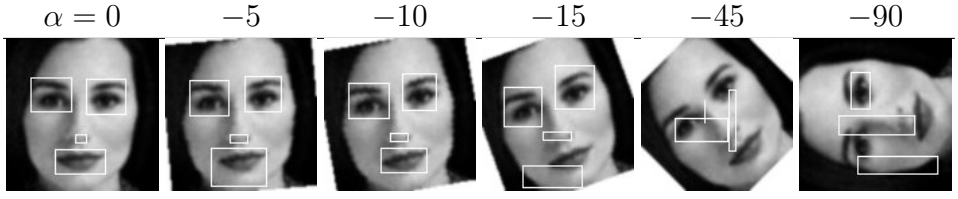

Figure 3.2: Face salient extraction results according to roll angle α.

In order to have a correct extraction, we need a roll variation less than 10 deg. Notice that we must make a difference between a correct extraction and a credible extraction. In Figure 3.2, the extraction of the mouth with a roll of 15 deg is credible but not correct. Studying the credibility of face elements can be time consuming and not enough accurate.

3.3.1.2 Global Horizontal energy

In order to have more accurate estimation of head roll, we will come back to the global horizontal energy. As a remainder, the horizontal energy uses Haar horizontal mirror pattern. It is expressed at each point of the image, thus, it can be seen as a local horizontal energy.

Figure 3.3 shows a Haar horizontal filter applied to the point A. The point "A" is on the straight line which separates a region of null intensities (black pixels) and another one of maximum intensities U (white pixels). β is the angle of rotation of center "A". A null rotation β corresponds to the configuration of a straight horizontal line (a).

The local horizontal energy is the absolute value of the difference between the sums of the upper and lower part of the filter. Since black regions have null intensities, we will only be interested in maximum intensities inside the Haar pattern, in Figure 3.3b, pixels of maximum intensities are represented in blue or green.

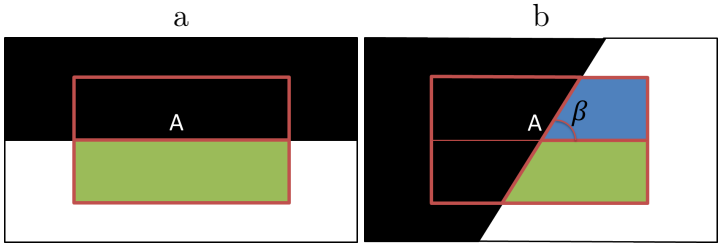

Figure 3.3: Image a illustrates a horizontal straight line with the horizontal Haar filter. Image b illustrates an oblique straight line with the horizontal Haar filter centered in A. In (a), the energy is the area of the green zone. In (b), the energy is the difference of the areas of the blue and green zones.

Geometrically, it is obvious the local energy has a maximum when the contour containing the point "A" is horizontal $\beta = 0[180]$. Since the pattern has 2 symmetry axis, given E_{Hh} the local energy, $E_{Hh}(A, \beta) = E_{Hh}(A, -\beta) = E_{Hh}(A, 180 - \beta)$. In other words, we need to study only rotation of the interval $[0, 90]$. Given Hh the Haar horizontal pattern of width w and height h, the equation 3.1 corresponds to E_{Hh} of point "A" with a rotation of β in the interval $[0, 90]$.

$$\begin{aligned} E_{Hh}(A,\beta) &= U\left|\frac{1}{2}wh - \frac{1}{4}w^2\tan\beta\right|,\ \beta \in \left[0, \arctan\frac{h}{w}\right] \\ E_{Hh}(A,\beta) &= U\left|\frac{h^2}{4\tan\beta}\right|,\ \beta \in \left]\arctan\frac{h}{w}, 90\right[\\ E_{Hh}(A,\beta) &= 0,\ \beta = 90 \end{aligned} \tag{3.1}$$

The equation 3.1 shows that there is a unique maximum for $\beta = 0\,\text{deg}$ and $E_{Hh}(A, 0) = \frac{1}{2}wh$. So, each pixel close to a horizontal contour has a high local horizontal energy. As the head roll is close to a null roll, face elements, nose basis, eyes, and mouth will have approximately a horizontal direction. Hence, local energy of almost all pixels in salient regions should generally be high. Other parts of the face are more homogeneous and should have a low local energy, whatever the roll is.

A global horizontal energy E_G is then defined as the sum of all local horizontal energy of pixels included in the face. However, face segmentation is a difficult task and will be time consuming. To limit the face, we propose a circle defining the face position. It will enable to highlight face elements. The center of this circle is the center of the square window (Figure 3.4a). This mask Ω will be simple to generate and to apply when we will compute the global energy. Moreover, since we assume face detection has succeeded,

this circular mask should contain all the face salient elements. Roll of the face will be defined as the rotation the center of which is also the center of the face window. Hence, whatever the roll value is, the image on which the mask is applied, should contain the same set of pixels (Figure 3.4b and 3.4c).

One can wonder why we do not use the face mask described in 2.4.3. There are two reasons for that:

1. The main reason is that the vertical borders require an approximate null roll face. Indeed, for example, when face roll values is $\pm 90\,\text{deg}$, face main boundaries will be lines of approximate horizontal direction, but our method of face mask extraction uses approximate vertical lines to detect the mask, and thus it is not be able to generate the face mask.

2. Another reason is that we do not need a region which includes all the eyes, nose and mouth. Actually, a region which includes parts of the eyes, nose and mouth should be sufficient to estimate the roll.

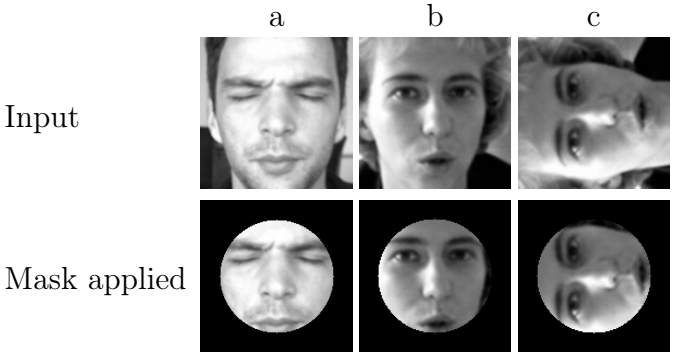

Figure 3.4: Original face window and application of the circular mask on the face window.

Given E_{Hh}, the local energy, given Ω the circular mask, and $Ro(\alpha, \Omega, I)$ the rotation operator of angle α using the mask Ω on the face window I, the global energy E_G according to the rotation α is defined by the equation 3.2.

$$E_G(\alpha) = \iint_{Ro(\alpha,\Omega,I)} E_{Hh}(x, y)\, dx\, dy \tag{3.2}$$

The horizontal global energy should be maximal with a null face roll value. Let us start by the graph of global horizontal energy of Figure 3.4a according to the rotation α.

Since the local and hence the global energies are symmetric for a rotation of 180 deg, only a rotation between $[-90, 90]$ will be taken into account. This image has a roll value very close to zero. Hence, we should find a maximum on the graph for a rotation of zero degree. Figure 3.5 shows this graph associated with this image of Figure 3.4a.

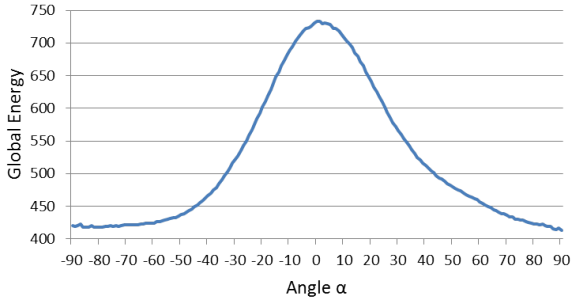

Figure 3.5: Graph of the global horizontal energy of Figure 3.4a according to the rotation α. In the original image, the roll of the face is almost null.

As expected, a maximum of the energy is found for a null roll value. On the other hand, one can notice that the graph is piecewise monotone.

Now, let us see the similar graph associated with the images of Figure 3.4b. In this image the roll is also almost null, so we should also find a maximum value of the global horizontal energy for a null rotation. However, as we can see in Figure 3.4c on which a rotation of -90 deg is applied (the ordinate axis is oriented from the top to the bottom, a positive rotation will be clockwise), face is illuminated from the left. With this rotation, in Figure 3.4c, some vertical lines appear, essentially located on the nose and on the cheeks. Therefore, the global energy graph according to the rotation α should present a maximum value for a null rotation, as well as another local maximum value for a rotation of $\alpha = -90$ deg. Since the global energy value should be the same with a rotation of 180 deg, it also means that we should have a third local maximum around the rotation value $\alpha = 90$ deg. Figure 3.6 shows this graph.

As expected, three local maximums are observed. The global maximum value of the global horizontal energy is reached for a rotation of 85[180] deg, despite the roll of the face is null. So, in order to estimate the roll, finding the global maximum of the horizontal energy graph will not be enough. On the other hand, we can obviously assume that the roll will be estimated

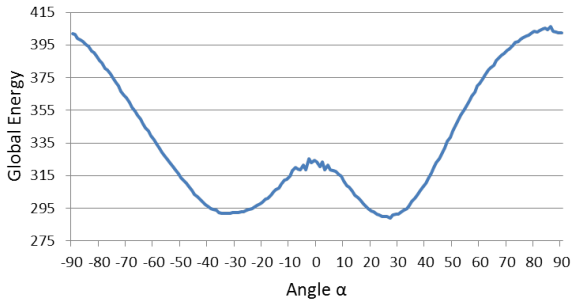

Figure 3.6: Graph of the global horizontal energy of Figure 3.4b according to the rotation α. In the original image, the roll of the face is almost null.

by the rotation corresponding to a local maximum. Contrary to the roll estimation using only the face salient elements credibility, the roll can be estimated quite accurately depending on the discretization of possible angles. Here, we choose a discretization of 1 deg. However, all we can know from the global energy according to the rotation α is that the roll value of the face should be one of the rotation α_i where $E_G(\alpha_i)$ is a local maximum. In other words, the study of the global energy variation will give a set of possible roll values. To be exploitable, the number of local maximums and their associate rotation angle should be low. These properties will be considered to fix the global strategy we present in next section.

3.3.1.3 Global Scheme of roll estimation method

Given a face window I, we, first, extract local maximums of the global horizontal energy according to the rotation α. We assume that local maximums should be generally separated by a quite large rotation α. We compute the global energy only every 10 deg from -90 deg to 90 deg. From these 19 values of global energies at different discrete rotation values α, variation of the global energy as well as the interval of rotation angle $[\gamma_i, \gamma_i + 20]$ in which there is a local maximum are computed. Then for each interval $[\gamma_i, \gamma_i + 20]$ the local maximum of global horizontal energy $E_G(\alpha_i)$ will be the maximum value of E_G in the interval $[\gamma_i, \gamma_i + 20]$. As a result, we will get the set $\Theta = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ of the n candidate rotation angles of the face roll corresponding to each of the n local maximums $E_G(\alpha_1), E_G(\alpha_2), \cdots, E_G(\alpha_n)$.

The second step consists of computing the accurate rotation angle within Θ. We will use the face salient element extraction method. We assume that local maximums are separated by a quite significant angle. For each α_i of Θ, we study the credibility of extracted salient elements to build a credibility score. The extracted elements of a specific α_i with

the highest credibility score will correspond to the elements of a null roll. Figure 3.7 shows the global scheme of the roll estimation method.

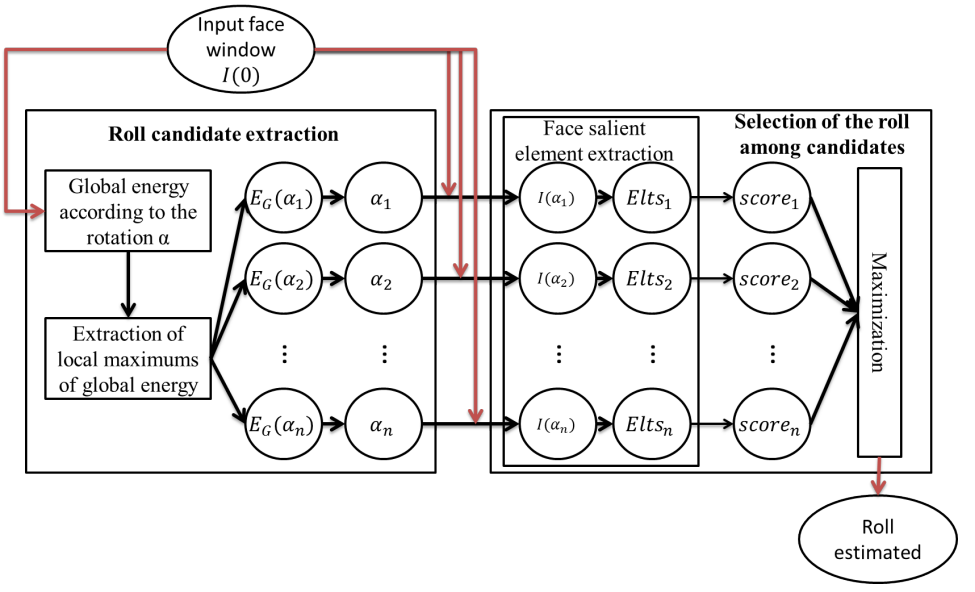

Figure 3.7: Global scheme of the roll estimation method.

The next paragraph will now focus on how we select a candidate rotation among the possible values.

3.3.1.4 Roll estimation by selecting among candidate rotation

As said before, we have at this moment a set $\Theta = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ of possible roll values for a given face. For each of the possible candidate roll values α_i, a rotation of α degree has been applied on the original face window. Then, we extract salient features on each rotated face image. The face salient elements extraction method localizes elements in their bounding boxes. However, either we choose the final result with only four boxes (right and left eyes, nose and mouth) using the multi-threshold analysis or we choose all the bounding boxes extracted from all the 18 binarized energy maps, each of them obtained by a discrete threshold value.

In the first case, the only credibility of each of the four extracted elements will probably be not enough to select a candidate rotation for face roll estimation. Since there are only four boxes, the information about the relative positions and sizes lack of accuracy. Further features on each element must be extracted. On the other hand, if we use the 18 thresholds which generate 18 energy maps, 4 face elements, at most, being extracted from each map, we could have at most 72 face elements extracted for each

rotated face window. We do not want to simply multiply face elements to have redundant information, but we also want to stress the stability of extraction method for low variation of the threshold. Then, extracted elements should be more stable despite the variation of the threshold in a null roll face than the elements of a rotated face. Figure 3.8 shows the salient element extraction according to the threshold of energy map binarization and the rotation angle α related to Figure 2.31.

As expected, relative positions and sizes of face elements with a null roll face windows are more credible than those from rotated views of the face window. Moreover, face elements positions and sizes are more stable despite the binarization threshold variation. Therefore, we assume that the analysis of this stability as well as the analysis of the credibility of each region should produce a robust score system. The score is built by incrementation, accumulating the evidence of a good element extraction. Then, it will be upper bounded by 72, the number of regions extracted using 18 different thresholds.

A face salient region of a given threshold which satisfies some criteria will add a point to the score SC which will depend on the rotation candidate angle α_i. Hence, for each candidate rotation $\alpha_i \in \Theta$, the score gives the number of face elements which are credible and a large score is an indication of stability. α^*, the angle value giving the highest score is the estimated roll, α^* is also the candidate rotation which maximizes the score as shown in equation 3.3.

$$- \alpha^* \in \Theta / SC(\alpha^*) = \underset{\alpha_i \in \Theta}{max} SC(\alpha_i) \tag{3.3}$$

As we said, some criteria must be satisfied by each region. Some of them concern all the extracted elements, others are specific to a face element and others concern two face elements. Here are a list of these criteria.

1. Whenever a region is extracted, it must be significant. Therefore the bounding box area of the extracted element should be greater than 1% of the face window area.

2. A null roll face region is composed of approximate horizontal lines. Moreover, face elements themselves are more horizontal than vertical. The width of element bounding box must be greater than its height.

3. We assume the whole face is detected. Therefore, an eye cannot have a width which would be greater than half of the window width. Hence, an eye region bounding box must have a width less than half of the face window width.

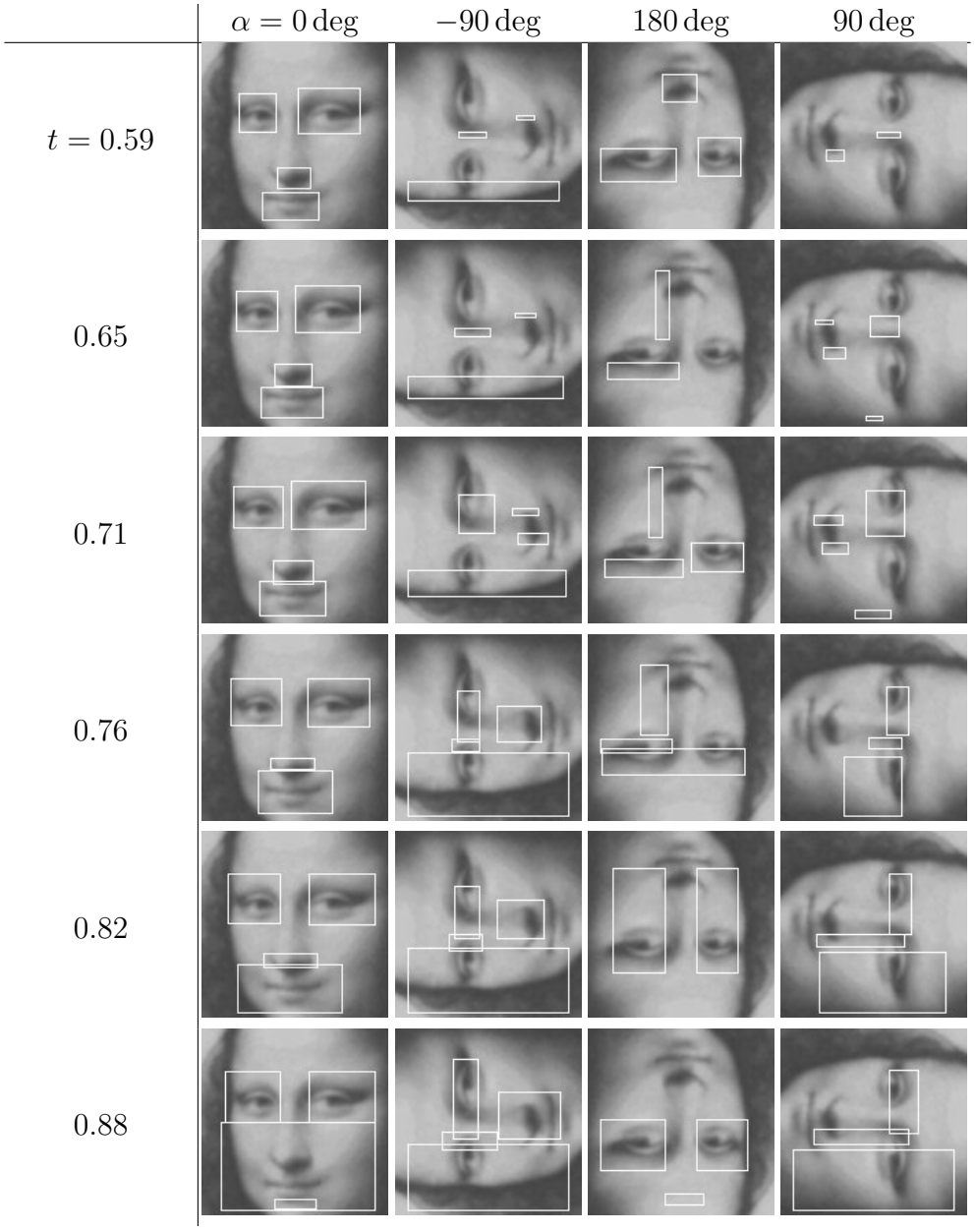

Figure 3.8: Some bounding boxes of extracted salient face elements according to the threshold applied on local horizontal energy map and the rotation angle α.

4. For similar reasons, a nose or mouth must have a width less than 60% of the face window width.

5. In a null roll face, both eyes must be almost at the same ordinate. In other words, when we project each eye region on y-axis, the inter-

section of the projections must not be empty.

6. Similarly, in a null roll face, mouth and nose intersect the face symmetry axis. In other words, when we project the mouth and nose on x-axis, the intersection of the projections must not be empty.

All candidate regions (at most 72 regions) are evaluated using these six criteria.

- Each time an eye region satisfies the criteria 1, 2 and 3, a point will be added to the score SC.
- To add a point to the score SC, a mouth or a nose must satisfy the criteria 1, 2 and 4.

However, when the criterion 5 is not satisfied, it means that both eyes exist but their spatial distribution is not possible for a null roll face. Similarly, when the criterion 6 is not satisfied, it means that nose and mouth were extracted, but they are not aligned and thus are in an impossible spatial distribution in a null roll face. In both cases, we cannot be sure that both regions (the eyes, or the nose and mouth) are wrong. However, we can obviously assume that at least one of them is wrong. Therefore, when criteria 5 or 6 are not respected, a point is subtracted from the score SC.

Finally, each candidate rotation α_i is associated with a score $SC(\alpha_i)$, the roll estimation will be the candidate rotation $-\alpha_i$ which is the maximum among all computed scores.

Now we presented a method which estimates the roll, the next section will present a method of yaw and pitch estimation.

3.3.2 Yaw and pitch estimation

Here, a geometric method which estimates the yaw and the pitch is presented. This method is based on the spatial distribution of the extracted region bounding boxes [Pyun et al., 2014b]. Since the face element extraction method requires an approximate null roll face window as the input, this estimation method needs a face with an approximate null roll. Here, the yaw and the pitch are estimated independently. However, the general steps of these estimations approaches are the same. We know that some sizes or relative positions of these elements should respect some basic knowledge on human face. This approach is close to how human vision

perception works to estimate the pose. Indeed, human vision estimates head pose using some cues on eyes, nose and mouth sizes and positions. Human vision is not able to estimate the pose accurately, except for some extreme poses like frontal and profile face views. As a reminder, the aim of this thesis is extracting one or a few of the most representative faces in order to recognize more easily the face in a video. In this part, we do not need to estimate the yaw or pitch with accuracy for all values, but we want the method to be fast and to present a better recall and precision as the face is frontal. Therefore, a geometric method is appropriate. Geometric methods are known for their rapidity and we will show they can achieve good estimation, in particular with frontal faces.

3.3.2.1 Motivation

First, to estimate the yaw, we assume that some parts of the faces have the same widths in the 3D real world. Left and right eyes must have almost the same widths. Mouth and nose basis are located on face symmetry axis. The distance of the left corner of nose basis and mouth and the distance between the right corner of nose basis and mouth should be the same. Let's consider the eyes. We assume that the eyes are located on the surface of a cylinder of radius R. Then the yaw rotation axis and the revolution axis of the cylinder are the same. Let us consider a single eye which has an angular width of w_0. All plan that intersects the cylinder perpendicularly to the revolution axis will be a circle of radius R and the center of which belongs to the revolution axis. An image is a projection of the 3D object on the image plan. Figure 3.9 shows a schematic view of such a configuration.

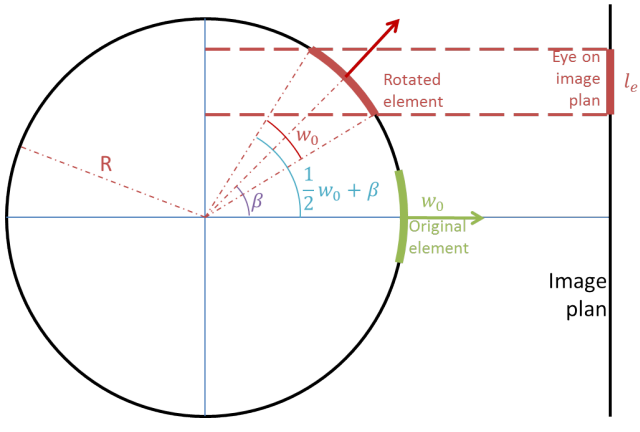

Figure 3.9: Schematic view of eye image plan projection.

Given β the yaw rotation, we can see that if $\beta \in \left[90 - \frac{w_0}{2}, 90 + \frac{w_0}{2}\right]$ or $\beta \in \left[-90 - \frac{w_0}{2}, -90 + \frac{w_0}{2}\right]$, a part of the eye will no longer be visible

from the image plan. If $\beta \in \left]90 + \frac{w_0}{2}, 270 - \frac{w_0}{2}\right[$, the eye will be entirely occluded. The length l_e of the eye in the image plan is computed by the equation 3.4.

$$\begin{cases} l_e &= 2R\left|\cos(\beta)\sin\left(\frac{w_0}{2}\right)\right| & \text{if } \beta \in \left[-90 + \frac{w_0}{2}, 90 - \frac{w_0}{2}\right] \\ l_e &= R\left|1 - \sin\left(\beta - \frac{w_0}{2}\right)\right| & \text{if } \beta \in \left]90 - \frac{w_0}{2}, 90 + \frac{w_0}{2}\right] \\ l_e &= R\left|-1 - \sin\left(\beta + \frac{w_0}{2}\right)\right| & \text{if } \beta \in \left[-90 - \frac{w_0}{2}, -90 + \frac{w_0}{2}\right[\\ l_e &= 0 & \text{if } \beta \in \left]90 + \frac{w_0}{2}, 270 - \frac{w_0}{2}\right[\end{cases} \tag{3.4}$$

We can prove that a maximum of l_e is reached for the rotation $\beta_{max} = 0$. Hence, the projection length or an element width projection will be maximum when the element is at the central position of the face. As the element is far from a null yaw position, the width should decrease.

Now we study the impact of the projection on widths of facial elements, we can introduce other knowledge to estimate geometrically the yaw. Indeed, eyes have the same widths. So, if the projections of two eyes on the image plan have the same width, it means that, if an eye has a rotation angle of $-\beta$, the other one has a rotation of β, these eyes are symmetric in the image plan and thus, the face has a null yaw value.

On the other hand, when there is a difference between the widths of left and right eyes on image plan, it means that the head is turning in the same direction as the place where the less wide eye is. As we see in Figure 3.9, it is possible to compute a continuous estimation of the yaw with two eyes widths when we know the angular width separating the left and right inner endpoints of both eyes. Unfortunately, despite we can say that eyes have almost the same angular width, the angular width of the inner endpoints of both eyes depends on the considered face.

However, we will consider that, generally, the angular widths between both eyes are almost the same as the angular width of the eye. At least, we know these distances are in the same order of magnitude. Moreover, when faces are almost profile views, the nose bridge will hide a part of one eye. The width expression of projected eye will not be exactly as we expressed in formula 3.4.

About the pitch, relative positions of face elements are different from a face to another. Moreover, when faces are oriented downwards or upwards, elements are merging and cannot be separated.

Generally, despite a few limitations, knowing the positions and sizes of face elements will give cues to estimate the yaw and the pitch of the head. Hence, we first use the face salient element extraction method. Now we

have the bounding boxes of eyes, nose basis and mouth, we can use some knowledge on face distribution to estimate the yaw and the pitch.

3.3.2.2 Estimation intervals of yaw and pitch

Here, a left element will be the one on the left part of the face. In other words, "left" or "right" are relative to the image plan and not to the 3D face object. From previous remarks, we know that such geometric method, with such configuration cannot give accurate estimations of the yaw, especially for faces of almost profile views. Therefore, we will estimate only some intensities of yaw. We consider only nine discrete values represented by the set $S_Y = \{-4, -3, \cdots, 0, \cdots, 4\}$ where:

- the element -4 represents left profile view,
- the element -3 represents almost left profile view,
- the element -2 represents left diagonal view,
- the element -1 represents almost left frontal view,
- the element 0 represents a null yaw view,
- the element 1 represents almost right frontal view,
- the element 2 represents right diagonal view,
- the element 3 represents almost right profile view,
- the element 4 represents right profile view of the face.

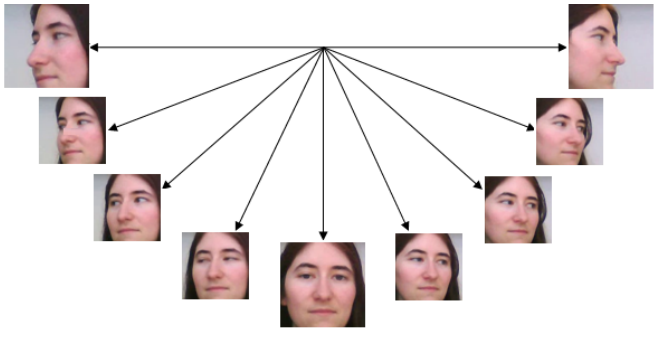

Figure 3.10: Examples of faces depending on the yaw interval.

Figure 3.10 shows examples of faces depending on the yaw interval. Similarly, the pitch is represented by 3 discrete values designated by elements of the set $S_P = \{-1, 0, 1\}$ where:

- the element -1 represents a downward view,
- the element 0 represents an almost null pitch view,
- the element 1 represents an upward view of the face.

Now we have defined how are the discrete estimation angles of face yaw and pitch, we will first start with presenting the yaw estimation method.

3.3.2.3 Estimation of head yaw

As a reminder, face salient elements (eyes, nose basis and mouth) are extracted in their bounding boxes. As all faces are at the same scale as the face window, all positions and sizes are normalized by the length L of the square face window. Therefore all abscissas, ordinates, widths and heights in the bounding boxes have a value in the interval $[0, 1]$. Figure 3.11 shows schematic representation of these bounding boxes in left profile and diagonal views, in null yaw view and in right diagonal and profile views.

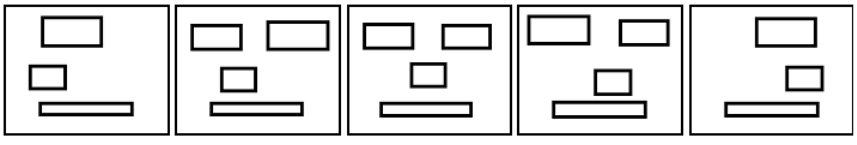

Figure 3.11: Schematic representation of the extracted bounding boxes in left profile and diagonal views, in null yaw view and in right diagonal and profile views.

A yaw variation has a significant impact on element widths as well as on the relative positions, especially the element abscissas. As the face turns on the right, the right eye will be smaller whereas the left one will be larger. As a result, the difference of the eye widths should increase. Moreover, if the left eye is smaller than the right eye, it can be associated with a yaw towards the right. Otherwise, the yaw will be oriented to the left.

Besides, nose and mouth relative positions vary too. When face yaw value is null, nose and mouth are aligned in their centers. If the face turns to its right, the nose will shift to the right. Otherwise, the nose will shift to the left part of the mouth.

These observations were translated into two functions. First, the function E will concern the eyes widths, whereas the function N will concern the relative positions on mouth and nose.

First each element are projected on x-axis. d_L is the projected width of left eye whereas d_R is the projected width of the right eye. Nose and mouth are also projected on x-axis. δ_L is the abscissa difference between the projections of the nose and mouth left corners whereas δ_L is the abscissa difference between the projections of the mouth and nose right corners as shown in Figure 3.12.

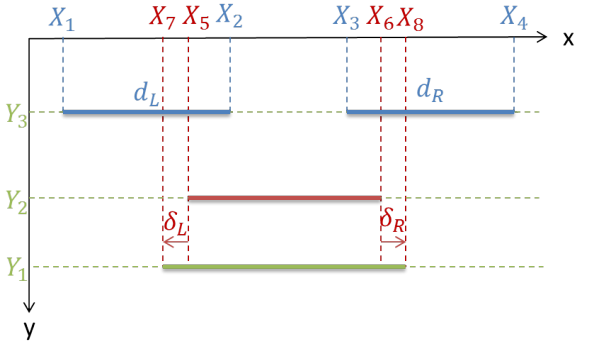

Figure 3.12: Parameters extracted from face salient element bounding boxes.

E is a function using the eye widths. First, we define d, the difference between the left and right widths as well as s which is 1 if $d_L \geq d_R$ and is -1 if $d_R > d_L$. When only one eye is visible in the image plan, we consider that $d = +\infty$. The equation 3.5 gives the expressions of d and s.

$$\begin{array}{rcl} d & = & |d_L - d_R| \\ s & = & \begin{cases} \frac{d}{d_L - d_R} & \text{if} \quad d_L \neq d_R \\ 1 & \text{if} \quad d_L = d_R \end{cases} \end{array} \tag{3.5}$$

d will give a cue on how far is the face yaw compared to a null face yaw whereas the sign of s will indicate if it is left or right rotation. Given ϵ and C_E two positive constants, the function E relative to eyes is defined by 3.6.

$$\begin{cases} E = e_0 & \text{si} \quad d \in [0, \epsilon] \\ E = s \times e_1 & \text{si} \quad d \in]\epsilon, C_E] \\ E = s \times e_2 & \text{si} \quad d \in]C_E, +\infty[\\ E = s \times e_3 & \text{si} \quad d = +\infty \end{cases} \tag{3.6}$$

If $E = e_0$, it means that we have a frontal view, since the difference between the left and right eye widths is almost zero. On the other hand if $E = \pm e_3$, it means that we should have a profile view, since only one eye is visible.

Similarly, the function N gives another cue about the yaw estimation using relative position of nose and mouth. Let us define l the absolute value of the difference between δ_L and δ_R and s_N the function which gives whether the face is turned to the right or to the left (equation 3.7).

$$\begin{array}{rcl} l & = & ||\delta_L| - |\delta_R|| \\ s_N & = & \begin{cases} -1 & \text{if} \quad \delta_L \leq \delta_R \\ 1 & \text{if} \quad \delta_L > \delta_R \end{cases} \end{array} \tag{3.7}$$

The function N is then expressed by the equation 3.8 where C_N is a constant.

$$\begin{cases} N = -n_2 & \text{si} \quad \delta_L > 0 \\ N = n_2 & \text{si} \quad \delta_R < 0 \\ N = s_N \times n_1 & \text{si} \quad l \in]C_n, +\infty[\\ N = n_0 & \text{si} \quad l \in [0, C_n] \end{cases} \tag{3.8}$$

The first two expressions of equation 3.8 are first tested before the last two expressions. $N = \pm n_2$ should happen with profile or almost profile views, since it means, at least a part of the nose projection on x-axis does not intersect the mouth projection on x-axis. The face symmetry axis moves to the left ($-n_2$) or to the right (n_2). On the other hand when $N = n_0$, it means that nose and mouth are almost aligned on the face symmetry axis. Such configuration happens with a null yaw face.

The function E gives 7 different values and the function N gives 5 different values. In order to estimate the yaw, a combination of E and N gives the final estimation of the yaw. Table 3.1 shows the yaw estimation rules.

Table 3.1: Yaw estimation according to E and to N. "ND" means "not determined".

N & E	$-e_3$	$-e_2$	$-e_1$	e_0	e_1	e_2	e_3
$-n_2$	-4	-3	-2	ND	ND	ND	ND
$-n_1$	-4	-3	-1	ND	ND	ND	ND
n_0	ND	ND	0	0	0	ND	ND
n_1	ND	ND	ND	ND	1	3	4
n_2	ND	ND	ND	ND	2	3	4

In the table 3.1, some combinations of E and N values are not realistic. For example, $E = e_0$ suggests that eyes have almost the same width. This happens with almost null yaw face. However if $N = -n_2$ at the same time,

it means that nose and mouth relative positions suggest an approximate left profile view. Hence, they are inconsistent, the yaw cannot be estimated.

3.3.2.4 Estimation of head pitch

Estimating the pitch using bounding boxes of salient regions is a difficult task for two reasons. First, when face looks upwards or downwards, elements tend to merge. In particular, when a face looks downwards, nose bounding box can be entirely included in mouth bounding box. So the extraction itself will often fail. Second, even when the face elements are well extracted, bounding boxes configuration of face with extreme positive or negative pitch will look like each other. So, it will be difficult to evaluate correctly the pitch with such geometric method.

However, it is still interesting to see the evaluation of the pitch estimation with this geometric method. Let us define the highest ordinate Y_E of bounding eye bounding boxes. Since the origin in the face window is the left upper corner of the window, given y_{LE} and y_{RE} respectively the ordinates of the upper left vertex of left and right eye bounding boxes, Y_E is defined by the equation 3.9.

$$Y_E = min\left(y_{LE}, y_{RE}\right) \tag{3.9}$$

Let us call Y_N and Y_M respectively the ordinate of the nose basis and the ordinate of the mouth. When a face is looking upwards, the distance $\delta_{up} = |Y_E - Y_N|$ between the eyes and nose decreases while the distance $\delta_{dn} = |Y_N - Y_M|$ between the nose and the mouth is almost the same. On the other hand, when a face looks downwards, δ_{dn} should be less than the distance δ_{up} between eyes and nose. Therefore, the pitch P can be evaluated by the equation 3.10.

$$\begin{cases} P = -1 & \text{if } \delta_{dn} < \frac{1}{2}\delta_{up} \\ P = 1 & \text{if } \delta_{up} < \frac{1}{2}\delta_{dn} \\ P = 0 & \text{otherwise} \end{cases} \tag{3.10}$$

We choose a geometric method to estimate the pose for two reasons. First, we want to show it can be a possible direct application of the previous extraction method. Second, in order to find a good face window candidate, we need to estimate the face pose. Indeed, many face recognition methods require to know the pose before processing. Indeed, knowing the face pose means that we can eventually correct the pose, in particular the roll, on

face window to get a frontal view. The next section will discuss on our face pose estimation method efficiency.

3.4 Evaluation of our pose estimation method

Here, we will present the evaluation of the pose estimation method. As a good extraction of face salient elements is required to estimate the yaw and the pitch and since an almost null roll value is needed to achieve the elements extraction, we will first evaluate our roll estimation method before the evaluations of yaw and pitch estimation method.

3.4.1 Evaluation of the roll estimation method

Roll estimation method will be evaluated on Color Feret and BioID. The roll of Color Feret images is labeled whereas in BioID, the roll is not labeled. However, all faces in BioID should have an almost null roll. Hence, for BioID, the roll will be considered as null. Notice that, when we estimate the roll, all rotations of the face are considered as a possible roll value. So, it does not matter that the real face roll is null or not. If we apply a rotation of $-\alpha$ on this image, our method will find that the roll value of this rotated image is α. In other words, there is no assumption about the roll of a face image at the beginning. On Color Feret and BioID images, many faces wear glasses or have mustaches or beards.

Traditionally, when we estimate a pose, whatever the orientation is, we compute the mean absolute error and the standard deviation of this error. However, these measures are not sufficient to properly evaluate the roll estimation. Indeed, roll estimation will be used as the first step of many applications on faces. Therefore, we should be interested in the ratio of the face images with an operable roll. A lot of algorithms of face recognition or tracking have a minimum roll operable error. In other words, if a roll is greater than this operable error, recognition or tracking will fail.

Given α_R and α_E respectively the real and estimated rolls, the absolute error err_{roll} is defined by the equation 3.11.

$$err_{roll} = |\alpha_R - \alpha_E| \tag{3.11}$$

Obviously, the mean absolute error is important, but the ratio of correct images is more important to see if the method can be used as a preprocessing step. For example, in our face element extraction, eyes detection rate are almost the same when the faces have a roll less than 15 deg, in Viola and Jones face detector, the detection requires a face roll less than 15 deg. Hence, it is interesting to know the proportion of images where the absolute error is less than a reception threshold. We assume that a roll will

be correctly estimated and operable if the absolute error $err_{roll} < 7\,\text{deg}$. Hence, the correctness is the proportion of images where $err_{roll} < 7\,\text{deg}$.

3.4.1.1 Correctness according to the number of local maximums

Our roll estimation method estimates the roll by extracting the local maximums of the global energies according to the rotation angle α. One of these local maximums will be selected to estimate the roll. So, the number of local maximums to test must be as low as possible. Previously, we saw that face windows should have generally one or three local maximums. Table 3.2 shows the distribution of faces and the correctness according to the number of extracted local maximums in Color Feret database.

Table 3.2: Distribution of faces and correctness according to the number of extracted local maximums in Color Feret.

Local maximum numbers	1	2	3	4	5	all
Distribution (%)	76.12	9.16	10.42	1.97	0.33	100
Correctness (%)	99.59	87.72	96.05	86.11	83.33	97.80

The global detection correctness is 97.80%. As expected, most of the images generate one or two or three local maximums. In particular, 76% of images in Color Feret database generate a single maximum. Moreover, the correctness of images with one local maximum of the global energy is 99.59%. In all Color Feret database, a face image can generate at most five local maximums. Hence, there are at most five candidate angles to test for the roll estimation.

3.4.1.2 Correctness according to the score

In our roll evaluation method, once candidate rotations angles α_i are found, with each local maximum of the global horizontal energy, a rotation angle α_i is associated. A rotation of α_i is then applied on the original face window. Using the salient element extraction, a score $SC(\alpha_i)$ is computed for each candidate rotation angle. In order to understand the relationship between the score and the correctness, we define the difference operator Δ using the highest $SC(\alpha_{M1})$ and the second highest $SC(\alpha_{M2})$ scores, corresponding respectively to the candidate rotation angle α_{M1} and α_{M2}. However, when there is only one candidate angle (one maximum is extracted

on the global horizontal energy graph), we assume that $SC(\alpha_{M2}) = 0$. The difference operator Δ is defined by the equation 3.12.

$$\Delta = |SC(\alpha_{M1}) - SC(\alpha_{M2})| \tag{3.12}$$

Figure 3.13 shows the distribution of faces images in Color Feret according to the difference operator Δ. There are two parts in this graph. The left part is composed by almost all the images which are associated with at least two local maximums. The right part of the graph corresponds to images associated with one maximum. Generally, as the number of local maximums (which is also the number of candidate rotation angles) decreases, the difference operator increases. Since we saw that the correctness with only one maximum is the best one, the correctness should also increase when the differences operator increases.

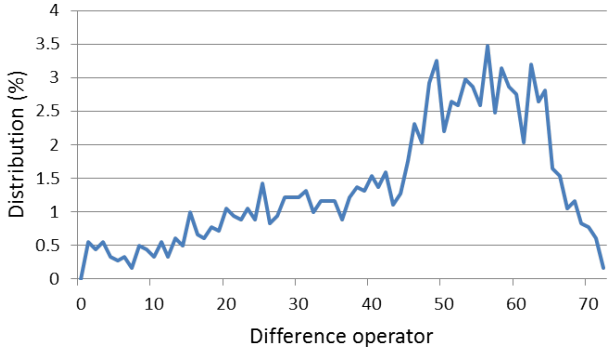

Figure 3.13: Distribution of face images (%) according to the difference operator Δ.

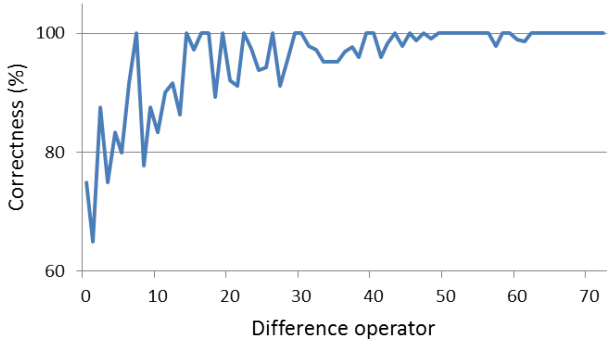

Figure 3.14: Correctness (%) according to Δ.

Figure 3.14 shows the graph of the correctness according to the difference operator Δ. As expected, with high values of Δ, the correctness is globally excellent whereas the correctness of low values of Δ is worse. However, the correctness according to the difference operator Δ is almost always greater than 80%.

To improve the correctness, we also propose a rejection rule. We observed that results are better when Δ is high. Thus, Δ can be used as a rejection threshold. Given the number of face windows with $\Delta < T$ and with $err_{roll} < 7\,\text{deg}$ $\sharp(T, err_{roll} < 7\,\text{deg})$ and given the number of face windows with $\Delta < T$ $\sharp(T)$, we define the correctness with rejection $C_R(T)$ by the equation 3.13.

$$C_R(T) = \frac{\sharp(T, err_{roll} < 7\,\text{deg})}{\sharp(T)} \tag{3.13}$$

Figure 3.15 shows the graph of the correctness according to the rejection threshold T of images of Color Feret database.

Figure 3.15: Correctness (%) according to the rejection threshold T.

3.4.1.3 Mean square error and standard deviation

We also evaluate the roll estimation method in BioID database and we obtain similar results. Table 3.3 shows the correctness, the mean absolute error and the standard deviation of the error in both Color Feret and BioID databases.

As we can see in table 3.3, the mean absolute error is around 4 deg in both databases. However, the standard deviation is high compared to the mean absolute error. These values involve two consequences.

Table 3.3: Correctness(%), mean absolute error and standard deviation of the roll estimation in Color Feret and BioID.

Database	Correctness (%)	Mean absolute error (deg)	Standard deviation (deg)
Color Feret	97.80	4.07	8.54
BioID	98.23	4.00	8.35

1. On the one hand, since standard deviation is high, some of the absolute errors of roll estimation must be very high.

2. On the other hand, since the absolute mean error is low, in most of the cases, roll estimations are quite accurate.

Indeed, the process of head roll estimation selects an angle among face rotation candidates which maximizes the score. These candidates are often separated by approximately 90 deg. Hence, when the good angle among face rotation candidates is selected, the roll estimation absolute error is generally lower than 4 deg. However, when the roll estimation fails, the absolute error should be generally around 90 deg.

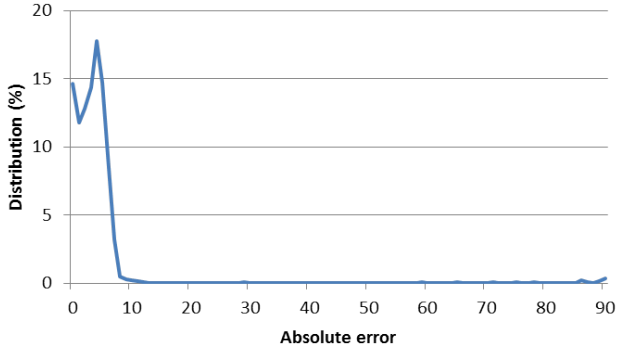

Figure 3.16: Distribution of face images (%) according to the absolute error in Color Feret database.

Figure 3.16 shows the distribution of face images according to the roll estimation absolute error. We are interested in high error values. However with such graph, it is difficult to see the distribution of high absolute errors. Figure 3.17 shows the logarithm of face images distribution according to the absolute error of roll estimation. As expected, most of the roll estimations are quite accurate. However, a few absolute errors are quite high. As a consequence, the standard deviation is also high.

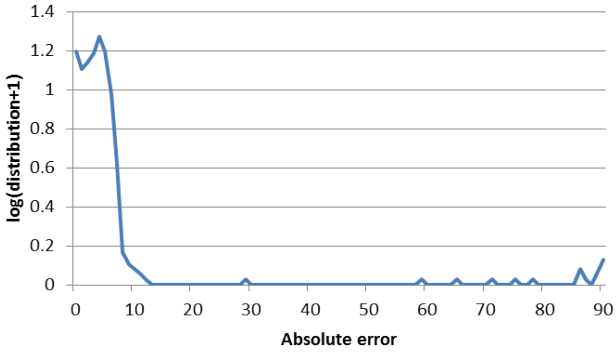

Figure 3.17: Logarithm of the distribution of face images according to the absolute error in Color Feret database.

3.4.2 Evaluation of yaw and pitch estimation method

Now the roll is estimated, we can easily apply a rotation in order to have an almost null roll value face. Salients elements have been extracted in their bounding boxes to estimate the yaw and the pitch.

3.4.3 Test database

To our knowledge, no labeled database exists to evaluate our method. In order to test the yaw and the pitch estimation method, we build a database of 566 images extracted from a video sequence with regular and continuous rotation of the head in both horizontal and vertical directions. Images yaw and pitch values are labeled manually with one of discrete poses. The yaw has one of the nine values of the set $S_Y = \{-4, \cdots, 4\}$ and the pitch has one of the 3 values of the set $S_P = \{-1, 0, 1\}$.

3.4.4 Parameters

Three parameters or constants are used to estimate the yaw:

- the parameters ϵ and C_E related to the difference between both eye widths.
- the parameter C_N related to the nose and mouth relative positions.

When we look at the equation 3.5, ϵ and C_E are some limits of the difference between left and right eye widths. In theory, it is difficult to

give the exact relationship between d and the corresponding yaw. Indeed, as we see in equation 3.4, the yaw β depends on the real width w_0 of the eye and the distance between both eyes. In 3D world, eyes are almost separated by a distance equivalent to the eye width. In theory, for a profile face the distance between the eyes will be infinite since only one eye is visible, a frontal face will have $d = 0$. In practice, we will consider that a face is frontal when this distance is less than a small value expressed here by the constant ϵ. On the other hand, when one of the eyes is still visible but is too small to be extracted, we will consider that it is a profile view. As we said, all coordinates of salient elements as well as their sizes are normalized to a value in the interval $[0, 1]$. Hence, all the constants should have a value in $[0, 1]$. Similarly to ϵ, C_N will discriminate an almost null yaw face from the others.

Table 3.4 shows the constant values used here as well as their approximate associated yaw estimation. The yaw estimation is not computed, we find these angles by analyzing the results on the images of our database.

Table 3.4: Constants value used for yaw estimation, as well as the experiments estimation of the yaw.

Value	$\epsilon = 0.15$	$C_E = 0.5$	$C_N = 0.40$
Yaw(deg)	10	60	25

Although it is not possible to give a continuous function which could associate a yaw with the difference of the eye widths d, since face spatial distribution is almost the same, the yaw associated with a constant value should not vary a lot. We can also understand why geometric method with such design cannot give accurate estimation of the yaw.

3.4.4.1 Evaluation of yaw estimation

In order to estimate yaw, we will first focus on the reliability of face salient elements detection despite the yaw variation. Then we can evaluate the yaw estimation.

Table 3.5 shows the element detection rate according to left yaw variation and rate of well classified left yaw per yaw class. A detection is considered as good, if all bounding boxes of both eyes nose and mouth are correctly found. As we can see, the extraction rate in this database is good. The detection of salient elements fails only with profile views. The extraction of salient face elements is reliable in this database. The yaw

Table 3.5: Element detection rate according to left yaw variation and rate of well classified left yaw per yaw class.

Yaw	Element detection rate (%)	Yaw estimation (%)
-4 (left profile)	77.78	77.78
-3	100.00	70.58
-2	100.00	55.56
-1	100.00	85.30
0 (frontal)	100.00	**95.90**

estimation rate is computed only on images where element extraction succeeds. We can see that estimation rates of frontal views are good compared to the others. As expected, the detection rate is better for frontal, almost frontal and profile views.

Besides, when the estimation fails, it is interesting to know how the estimation error is. This method estimates the yaw with respect to 9 poses. Therefore, to estimate the error, we must look at the confusion matrix (table 3.6).

Table 3.6: Confusion matrix of yaw estimation

Ground Truth/Estimated	-4	-3	-2	-1	0	1	2	3	4
-4	**7**	2							
-3	2	**12**	3						
-2		8	**15**	4					
-1			1	**29**	4				
0				2	**117**	3			
1					13	**20**	4		
2						5	**14**	2	
3							4	**16**	4
4							2	2	**31**

We can observe that when the yaw estimation fails, the error is almost always of one class. Hence, a failure can be explained by images close to both adjacent classes.

3.4.4.2 Evaluation of pitch estimation

As we said, when face looks upwards or downwards, the projections of elements should merge. Hence, element detection rate should decrease. Table 3.7 shows the element detection rate according to the pitch.

Table 3.7: Elements detection rate according to pitch

Pitch	Element detection rate (%)
1 (upwards)	62.83
0	100.00
-1(downwards)	70.40

As expected, element detection often fails with extreme values of pitch. Therefore, estimating the pitch will be difficult. With extreme pitch values, face elements are too close, but with our model we use the vertical distances between elements to estimate the pitch. Table 3.8 shows the confusion matrix of pitch estimation.

Table 3.8: Confustion matric of pitch estimation

Ground Truth/Estimated	1	0	-1
1	**47**	13	23
0	0	**117**	0
-1	25	35	**41**

As expected, pitch estimation is quite good with almost null pitch face where the recall is good. However, even for null pitch face, the precision is low. Indeed, many images where the face is oriented upwards or downwards are considered as frontal views. Moreover, as expected, the confusion matrix of pitch estimation shows also that our method cannot clearly discriminate upward face from downward face. Therefore, the elements extraction is not efficient for pitch estimation.

3.5 Conclusion of pose estimation

In this section, we presented two methods to estimate the head pose, according to the our goal. The first one estimates the roll, whereas the second one estimates the yaw and the pitch of the face.

Both methods use the face salient element detector presented in the previous chapter, but in different ways. In the roll estimation method, salient elements are used to validate a rotation as the face roll among some candidate rotations. In the yaw and pitch estimation method, the positions and sizes of element bounding boxes are the input of these estimations.

Despite both methods are geometric, the roll estimation method is able to give an accurate value, whatever is the face roll. On the other hand, the yaw and pitch estimation method gives only a quite wide interval as the estimations. However, the evaluation shows that the yaw, in particular, of frontal faces is well estimated. The accuracy of the estimation of approximate null yaw faces can be adapted by changing the value of a parameter ϵ. Indeed, a lower value of ϵ makes the approximate null yaw detection more selective.

We also show that salient elements extraction often fails with faces oriented upwards or downwards. Even when elements are correctly detected despite pitch variation, elements are too close, and spatial configurations of upwards and downwards faces are almost the same. Therefore, such geometric methods are not suited for pitch estimation.

Despite this limitation, remember that our pose estimation method can estimate the roll accurately. Moreover, the yaw estimation method is efficient with almost null yaw faces. In other words, we can detect almost null yaw faces. Moreover, the aim of this thesis is selecting the best face candidate image among frames of a video sequence. This pose estimation method is efficient to detect frontal faces (faces with roll, yaw and pitch null values).

Chapter 4

Face Tracking

Chapter contents

4.1 Introduction 125

4.2 Face tracking state of the art 126

4.2.1 Motion based methods 126

4.2.2 Model based approaches 128

4.3 Face Tracking Method 131

4.3.1 Finding the eyes at the first face sample 131

4.3.2 Tracking the region containing both eyes 132

4.3.3 Similarity functions 133

4.3.4 The search area 137

4.3.5 Selection of the best samples for the purpose of face recognition 138

4.4 Evaluation 144

4.4.1 YouTube Faces database 144

4.4.2 Choosing the similarity function 145

4.4.3 Tracking results 145

4.5 Conclusion 149

Chapter summary

In this chapter, we will present a new face tracking method based on the salient element extraction and its associated horizontal local energy map, as well as the pose estimation method.

This method will track only the region containing both eyes. We assume that this region is the one of the face regions which carries much information. In extension, from the tracking of the eyes, we will be able to track the whole face. In this chapter, we will also present how to extract some representative samples of this face in the video sequence.

Résumé du chapitre

Dans ce chapitre, nous présenterons une nouvelle méthode de tracking basée sur l'extraction des éléments saillants du visage et sur la carte d'énergie horizontale associée, ainsi que sur la méthode d'estimation de pose.

Cette méthode ne suivra que la région contenant les deux yeux. Nous supposons que cette région est l'une des régions renfermant le plus d'information. Par extension, à partir du suivi des yeux, nous serons capable de suivre le visage en entier. Dans ce chapitre, nous présenterons aussi comment extraire les échantillons représentatifs d'un visage présent dans une séquence vidéo.

4.1 Introduction

The previous chapter deals with the problem of extracting salient elements of a face and of estimating the pose. Given frames of a video sequence, we want to extract a few representative samples of all faces present in this sequence. Actually, giving as the output, a few representative samples of a face in a sequence has an interest only if we know where all the faces samples, even those which are not chosen, are at any moment of the sequence. In other words, we must track the faces in the video. Hence, face tracking will have two main benefits. First, while tracking the face, we should be able to evaluate some criteria to choose a few representative samples of faces. Second, the tracking itself will connect all the samples of the same face. Recognition of the representative samples will lead to the recognition of all these samples.

Tracking an object in the video consists in localizing this object, in all the frames. Trackers assume that the general appearance of this object does not vary a lot when we consider two consecutive frames. Indeed, two consecutive frames are two images of the same scene separated by a short lapse of time. Obviously, this assumption is still valid with faces. However, several kinds of variations can still change the object appearance:

- Illumination variation,
- Pose variation,
- Scale variation,
- Expression variation,
- Occlusion.

Face is a non rigid object and thus, it cannot be represented by a linear model. Moreover, video processing requires fast algorithms. Even when real-time is not required, the huge amount of images needs a moderate time complexity. Face tracking must also take into account this requirement.

The proposed face tracking method is based on the tracking of some features extracted from the horizontal local energy. Since eyes are the most salient elements of faces, we propose to track eyes instead of the whole face. Indeed, with the position and size of the eyes, when we know the pose, more exactly the face roll, we are able to find the whole face. The next section will present a state of art of face tracking approaches. Then, we will present our face tracking method and finally we will evaluate it.

4.2 Face tracking state of the art

In this section, we present a state of the art of face tracking methods. Face tracking methods can be divided into two main categories. First, the first category is composed of motion based approaches which try to track faces while extracting the face motion between two consecutive frames of a video sequence. Second, there are approaches based on a face model. These model based methods need first to build a face model. Using this model, they first extract the faces which fit this model in two consecutive frames in order to link them. The main differences between all these approaches are in the choice of the features to track and in the technique which will link faces of two consecutive frames.

4.2.1 Motion based methods

Motion based methods need to extract the face motion between two consecutive frames. They are either predictive or based on feature tracking or based on sequential detection.

4.2.1.1 Kalman filters

Kalman filter is widely used to estimate motion. In [McKenna et Gong, 1996], Kalman filter is used on disparity maps to predict the face motion. In [Comaniciu et al., 2000], the authors propose to use Kalman mean shift on color information of the area to track. It generates several target candidates. The target which maximizes the Bhattacharyya coefficients [Kailath, 1967] is assumed to be the most probable target. These methods require a good segmentation of the face, otherwise the motion prediction is influenced by the background.

In [Kim et al., 2007], the authors use Haar-like features to detect faces. A PCA is used to select the most representative features. Then a SVM is applied on these selected features to classify faces and non-faces. Kalman filter is then used on faces of current and next frames to track faces. In [Foyti et al., 2011], Kalman filter is still used to track faces. Modular Principal Component Analysis is then used to train on a database in order to recognize the faces. In [Zhu et Ji, 2004], the authors propose to estimate the pose and to track the face at the same time. The tracking system is based on Kalman filter. However, with large movements, the prediction fails. In order to improve the prediction, eye areas are introduced in the prediction process and enable to obtain more accurate results.

4.2.1.2 Particle Filter

Particle Filter gives a solution to hidden Markov Chain and nonlinear filtering problems. Given some partial observations, particle filter estimates the posterior state as a distribution of probabilities [Del Moral, 1995]. Therefore, particle filter can be used to predict the motion of the head in the next frame from a set of descriptors of the current frame. In [Stasiak et Pacut, 2007], the authors used the Conditional Density Propagation for Visual Tracking [Isard et Blake, 1998] in the particle filter. Face is detected using the skin color in HSV space. In [Hui et al., 2010], the authors use a color histogram and Local Binary Pattern histograms at different resolution as features of a particle filter. It shows that combined features can make the tracking more efficient. In [Yun et Guan, 2010], face fiducial points are tracked using multiple Differential Evolution Markov Chain [ter Braak et Vrugt, 2008] to combine multiple particle filters. Like other face tracking methods, particle filtering approaches cannot recover face tracking when it is lost. However, others try to overcome this. In [Mikami et al., 2009], the memory-based particle filter (M-PF) is proposed to track faces. The authors remove the Markov assumption which says that the predicted state depends only on the previous one. They try to include in the particle filtering all the previous target states. The prior distribution is generated from a random sampling of all previous states and then is associated with a new particle filter. They assume that such modifications make the tracking robust to brutal movements and occlusions, since the M-PF combines possible target states.

4.2.1.3 Active contours

Active contours [Lefèvre et Vincent, 2004] are also widely used in face tracking. A set of points is defined. Some forces inside the object tend to push these points to the outside, whereas other forces tend to push those points towards the inside of the object. An energy related to these forces is defined. When this energy is minimized, all the points should be on the contour of the object. In [Charoensak, 2004], active contour model is used to find face contour. Active contours are found for each face in each frame. The positions of the current points are used as the initial points for the next frame. Although active contour model is quite efficient to converge to a contour, initial position of points will have an important impact in the tracking process. First, if the face moves a lot, the points may converge to another part. Moreover, active contour model is time consuming. In [Huang et Su, 2004], the authors propose to reduce the search area using projection histogram of moving silhouette and a prior

face shape to make the tracking faster and more accurate. Despite these attempts, active contour model is still a time consuming process.

4.2.1.4 Wavelet Transform

Other approaches try to track face through its wavelet transform. The idea is to apply a wavelet transform on the face and hence the face is represented in another space, for example, the space of orientation frequencies. In [Kruger et al., 2000], Gabor wavelets are applied on faces. Then, faces of the next frame are matched in the wavelet subspace. Here, the authors proposed to minimize an energy function between the prior wavelets and the wavelets of another frame with several affine transforms (rotation and scale). However, these methods need to use many coefficients and wavelets and thus, are time consuming. In [Park et Lee, 2008], the authors assume that all Gabor wavelets are not relevant and therefore propose to learn the prior Gabor wavelets using Levenberg-Marquardt optimization method [Marquardt, 1963] and K-means clustering. These methods are robust to rotation and other affine transforms. However, they are also sensible to occlusions and are time consuming.

4.2.2 Model based approaches

Contrary to motion based methods, model based methods introduce high level knowledge, and thus, they guarantee that tracked frames fit a face model. However, they generally need more computation time. Indeed, they often need to apply some affine transform before tracking. Model based tracking methods require a face model. This model is generally based on skin, or on control point or on contour model.

4.2.2.1 Skin model approaches

Many researchers use human skin characteristics to detect or track faces in video. In [Kawato et Ohya, 2000], a model from skin color distribution is built as the faces are detected. A further step analyses the color histograms to track face. Similarly, in [Niu et al., 2003], skin color distribution is used to generate a statistical skin model. This step detects candiate face areas. A further step using some other facial features validates the face or not.

In [Destrero et al., 2007], a skin model is used to remove areas which do not contain face. Additional cues are introduced to detect face. The tracking itself uses a Kalman filtering. In [Vadakkepat et al., 2008], a

neural network separates the skin and non skin colors. Then, a skin color probability map is used to find face. The tracking can be proceeded using the Continuous Adaptive Mean Shift [Bradski, 1998]. In [Xia et al., 2006], frames are segmented using skin color information. The authors propose an adaptive skin based segmentation, they claim that this segmentation is more robust to lightning conditions. The main drawback with skin model tracking methods is that they require generally color images. Moreover, these methods are not robust to lighting variations.

4.2.2.2 Deformable model approaches

In these approaches, a deformable face model is required. Generally, a constant number of control points are used. Linking some of them may be an approximation of some facial features, such as face contours, eyes, nose and mouth. Elastic Graph is widely used. In [Stamou et al., 2005], a morphological elastic graph is proposed. Each vertex of the graph is labeled by a vector obtained by a multiscale dilation-erosion of the face image. The edges represent the relative positions of the vertices. Each vertex of the graph is located on a minimum of a cost function which computes a similarity measure between two consecutive frames, and hence tracks the faces.

Other researches use Active Shape Model (ASM) [Su et al., 2008] in face tracking. Shapes are represented as a set of control points. Each point is characterized by its gray level profile. From a training face database, a principal component analysis is then used in order to learn the possible deformation of this set according to the face pose. They also proposed a verification step; when a shape tracking fails, the Active Shape Model is restored with a new shape. This model needs reliable points. However, shape extraction can fail with a noisy background. Active Appearance Model (AAM) is an extension of Active Shape Model and is also used for face tracking [Kobayashi et al., 2008; Zhou et al., 2010c]. Each control point integrates, in addition, texture information. Active Appearance Model can produce impressive visual results because of an accurate face alignment. However, similarly to ASM, AAM often fails with cluttered background. In [Zhou et al., 2010b], the authors add some constraints during the matching step between two consecutive frames as well as a face segmentation in order to improve the tracker with cluttered backgrounds. ASM and AAM have also two main drawbacks. First, these methods are difficult to generalize. Face alignment often fails with new faces and if face exemplars of too many people are introduced during the training, very different face deformations will be considered by the Principal Component

Analysis and thus the alignment will fail too. Moreover, ASM and AAM consider only linear deformations. Therefore, when head is almost in a profile view, the face alignment will fail. To overcome this limitation, many approaches propose a multi-modal schemes [Cootes et Taylor, 1999; Grauman et Darrell, 2004]; they assume that face pose or expression variations can be described by a few linear statistical models. However, they are not able to handle both pose and expression variation. Other approaches try to overcome this limitation using a 3D face model [Xiao et al., 2004; Li et al., 2001]. Since a 3D model is built, these methods can handle in theory every new viewpoint. However, they are not robust to face expression variations and are time consuming. Finally, other researchers use nonlinear models [Sozou et al., 1995; Su et al., 2009]. However, these methods suffer from time complexity and therefore cannot be used in face video tracking.

Another remarkable face tracking approach [Krinidis et al., 2007] uses Scale-Invariant Feature Transform (SIFT). The first step in this method consists in selecting feature points, those which will be tracked. These points will then define a 3-D deformable surface model. Using this model, tracking points are evaluated accurately. In [Zhao et al., 2009], SIFT points are used to track faces. Homologous points of different face images extracted from different frames of a video sequence form a chain. Chains are then gathered and generate a spatio-temporal tube. These tubes are then used as features for actor retrieval in movies. On average, a face is tracked on 54 frames in 200 sequences of the french movie "L'Esquive".

4.3 Face Tracking Method

In this section, we will introduce a new method to track face in a video sequence. This face tracking approach is based on the extraction of face salient regions, then, it tracks these regions. However, Tracking all regions can be time consuming. Therefore, we only try to track the region containing both eyes. We assume that this region is particularly relevant and stable between video frames. Moreover, in [Juefei-Xu et al., 2011], the authors use Walsh-Hadamard transform encoded local binary patterns on the region containing both eyes for an age invariant face recognition. Despite the age variation, they obtain impressive results. It tends to prove that this region is particularly rich in information and, thus, is a suitable region to track in videos. We saw that most of the face tracking methods use either very local features (control points) or a global one (face skin) to track the face or some intermediate features like face contours. Our method does not require a high accuracy while tracking faces. Thus, we suppose that tracking face in the horizontal energy space could be sufficient. However, our face tracking method is not only global but it also uses intermediate features. Here, horizontal lines of eye region will be tracked.

Once a face is detected in a face window, the first step is to find the region containing one or both eyes. Tracking this region will be enough to track the whole face, since we are able to infer the whole face position and size, especially if we are able to know the face roll as it has been explained in section 3.3.1. The eye region is tracked in the next frame without performing a prior face extraction.

4.3.1 Finding the eyes at the first face sample

Given $Re(T)$ the rectangular region containing eyes in the frame T, to track this region, the idea is to find a rectangle in the search area of the next frame where the similarity of local energies is maximum.

As a reminder, eyes are detected using a horizontal local energy map. Given Hh the Haar-like horizontal filter, we use the energy map E_{Hh} (equation 2.11) defined in section 2.4.1. Then, the normalized energy En_{Hh} is computed by the equation 2.12.

As we saw, using En_{Hh}, the region containing the left eye and the one containing the right eye are extracted in their bounding boxes. The region Re containing both eyes is defined as the smallest rectangular region containing the previous bounding boxes. Given a region Z, we define the eye energy map E_Z^T of the region Z at frame T as the restriction of the

energy map En_{Hh} to the region Z (equation 4.1).

$$E_Z^T = En_{Hh/Z}(T) \tag{4.1}$$

4.3.2 Tracking the region containing both eyes

At frame $T-1$, the region containing both eyes $Re(T-1)$ is represented by the energy map E_{Re}^{T-1}. The aim here is to find the region containing both eyes $Re(T)$ at frame T. Given $Search(T)$ the search area at frame T, this region $Search(T)$ includes $Re(T-1)$ and its energy map is E_{Search}^T. For every point P in $Search(T)$, we define a rectangular region $S(P,T)$ where P is the upper left vertex of this region. The size of each region $S(P,T)$ is the same size as the previous region containing both eyes E_{Re}^{T-1} and its energy is $E_S^T(P)$. We assume the regions containing both eyes in two consecutive frames are very similar. Hence, the region $Re(T)$ should be the rectangular region $S(P^*,T)$ where a similarity function c is maximum. Figure 4.1 shows a scheme of the eye region tracking process based on similarity of energy maps.

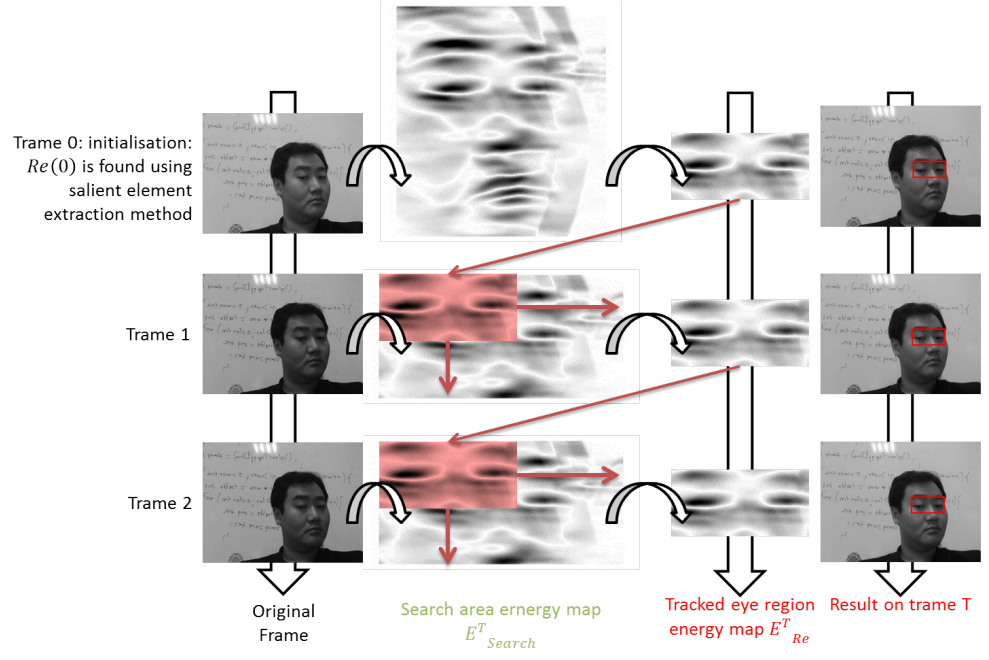

Figure 4.1: Scheme of the tracking process based on similarity of energy maps.

At the frame T, the similarity function c depends on the eye energy map of the previous frame E_{Re}^{T-1} as well as on the energy map $E_S^T(P)$. Hence,

the region containing both eyes $Re(T)$ at frame T and P^* the upper left point of $S(P^*, T)$ are defined by the equation 4.2.

$$\begin{aligned} Re(T) &= S(P^*, T) \text{ with} \\ P^* &\in Search(T) \text{ and} \\ c\left(E_{Re}^{T-1}, E_S^T(P^*)\right) &= \max_{P \in Search(T)} c\left(E_{Re}^{T-1}, E_S^T(P)\right) \end{aligned} \tag{4.2}$$

As a result, from the eye region $Re(T-1)$ of frame $T-1$, we are able to get a region $Re(T)$ in frame T. However, the scale can vary. It means that the size of $R_e(T)$ can be different from the size of $R_e(T-1)$. On the other hand, face roll can change. Thus, the similarity function c should take into account these possible variations. The question is how to compute the similarity function c.

4.3.3 Similarity functions

As we have already mentioned, the similarity function has to be accurate and with low computation time. It can be defined at different levels, according to the elements we consider in the energy map.

We have chosen to simplify the energy map, considering the binarized versions, but to keep information, we propose different binarization versions, associated, in our case, with 18 thresholds introduced in the section 2.4.4 about face salient element extraction.

In the first proposal, we compute at pixel level leading to accurate values but computation time becomes too long. Then, in the second proposal, we work at object level, the connected component level. The number of elements to compare is then reduced tremendously.

4.3.3.1 Similarity function at pixel level

The first similarity function uses the 18 binarized energy maps. Each of them uses one of the thresholds $\{t_1, t_2, \cdots, t_{18}\}$. Given a threshold t_i and a rectangular region Z, we define the binary energy map $B_Z^T(t_i)$ as the binarization of the energy map E_Z^T by the threshold t_i. Actually, the similarity function will depend on the similarity measures of binarized energy maps, as illustrated on Figure 4.2

The similarity function is defined as an aggregation of partial similarity functions computed with the same threshold t_i at times $T-1$ and T. We

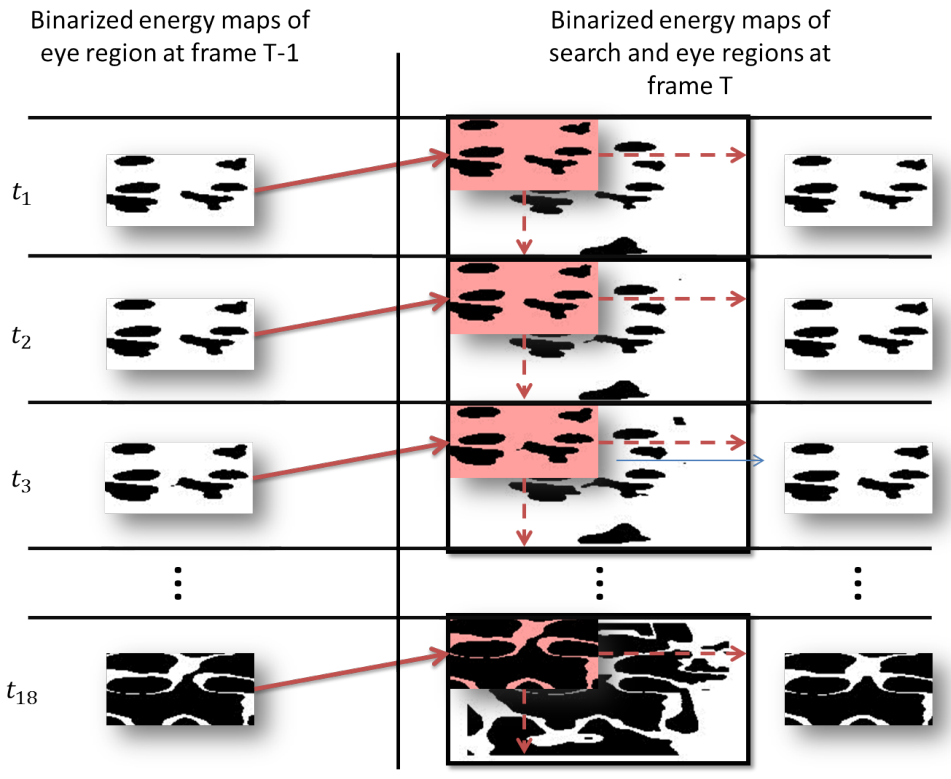

Figure 4.2: Scheme of the tracking process based on similarity of binarized energy maps.

define a partial similarity measure c_p between two energy maps binarized by the same threshold t_i. We use a binary correlation function between two zones of the same size (equation 4.3).

$$c_p(P,T,t_i) = corr(B_{Re}^{T-1}(t_i), B_S^T(P,t_i)) \tag{4.3}$$

c_p is associated with a fixed threshold value and this similarity measure computes actually the binary correlation between the binarized energy maps. From this partial measure of similarity, we can define a first similarity function c_1 at frame T, depending on the position of the point P of the region $S(P,T)$ by the equation 4.4

$$c_1(P,T) = \sum_{i=1}^{18} c_p(P,T,t_i) \tag{4.4}$$

Using $c_1(P,T)$ as the similarity function in equation 4.2, we are able to find the point P^* and the region containing both eyes $Re(T)$ in frame T.

However the computation time is quite expensive since the similarity function must be computed for each point P of the search area. In other words, the time complexity will depend on the size of the search area. This has motivated us to propose a second similarity function.

4.3.3.2 Similarity function at connected components level

The second similarity measure should overcome some of the drawback of the first function. Instead of computing a value for each point P of the search area, the second similarity will focus on the connected components of the binarized maps. As we said in a previous chapter, the binarized energy maps show the approximate horizontal lines of eye regions. In other words, the idea here is to track the horizontal lines in the region containing both eyes. They should be quite stable despite the variations which may appear between two consecutive frames. Skin based method need actually a time consuming segmentation step whereas control points method are often sensible to cluttered background. We assume that the horizontal components of eyes should achieve face tracking with a good accuracy and should be more robust to cluttered background than tracking method of control points.

From each binarized map $B_{Re}^{T-1}(t_i)$ of the eye region at frame $T-1$, bounding boxes of connected components are extracted in the set $J_i = \{u_1^i, u_2^i, \cdots, u_n^i\}$. Similarly, the energy map is computed in the whole search region $Search(T)$ at frame T. Then this search region energy map is binarized by each threshold $\{t_1, t_2, \cdots, t_{18}\}$. Bounding boxes of connected components of each binarized search region energy map are extracted in the set $K_i = \{v_1^i, v_2^i, \cdots, v_m^i\}$. The idea is to measure the similarity of n bounding boxes of the previous eye region with those present in the search area. The process will be faster, since the number of bounding boxes is much lower than the number of possible points P. Moreover, in the eye area, there are generally less than 20 bounding boxes.

Each connected component bounding box of the eye region at frame $T-1$ is matched with each connected component of the search area. For each couple (v_j^i, u_k^i), the other eye connected components of frame $T-1$ are matched with those from the search area of frame T. Since we consider only the bounding boxes, when a previous eye bounding box matches with a bounding box of the search area, it means that these rectangles intersects in a rectangle. The similarity measure is the sum of all common region (rectangles) areas. It can be seen as an approximation of the correlation measure. For each threshold t_i, the algorithm 1 returns the combination of bounding boxes extracted from the search area K_i which

matches the most with the bounding boxes extracted from the previous eye region J_i as well as the position of the previous eye region used for this matching. The similarity measure of a given combination is the common areas of previous eye region bounding boxes and search region bounding boxes. After applying this algorithm to each threshold t_i, since there are 18 different thresholds values, there are 18 different positions for the new detected eye region.

Algorithm 1 Returns the list of matching bounding boxes between the previous region containing eyes and the current search area, both binarized by the threshold t_i as well as the translation vector used for this best matching.

```
1: procedure List of matching bounding boxes
2:     listMax ← []
3:     maximum ← 0
4:     n ← 0
5:     vector ← null vector
6:     for each u_j^i ∈ J_i do
7:         for each v_k^i ∈ K_i do
8:             Match u_j^i with v_k^i
9:             tempVector ← translation vector applied on u_j^i for matching
10:            list ← []
11:            max ← 0
12:            for each u_l^i ∈ J_i do
13:                U ← translate u_l^i with tempVector
14:                for each v_ll^i ∈ K_i do
15:                    rect ← intersects U and v_ll^i (rect is a rectangle)
16:                    if area of rect is not null and is the maximum then
17:                        add v_ll^i to list
18:                        max ← max+ area of rect
19:            if max > maximum then
20:                listMax ← list
21:                maximum ← max
22:                vector ← tempVector
       return listMax, vector
```

Let us define $P = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, the positions of the new eye region found from each binarization threshold t_i. To compute the final position of the eye region at frame T, the mean of the abscissas m_x and the mean of the ordinates m_y are used. However, some of the values can be absurd. These values must not be taken into account while computing the means. Given $std(x)$ the absolute standard deviation of abscissas and

$std(y)$ the absolute standard deviation of ordinates. A position is selected if it respects the conditions 4.5.

$$|x_i - m_x| < std(x) \text{ and } |y_i - m_y| < std(y) \tag{4.5}$$

Given $P_S = \{(x_{S1}, y_{S1}), (x_{S2}, y_{S2}), \cdots, (x_{Sk}, y_{Sk})\}$ the set of positions which respect the conditions 4.5, we know that all points included in P_S should be quite close to each other. The final position (x_f, x_f) will simply be the mean position of all points of P_S as shown in equation 4.6.

$$\begin{aligned} x_f &= \frac{1}{card(P_S)} \sum_{i=1}^{k} x_{Si} \\ y_f &= \frac{1}{card(P_S)} \sum_{i=1}^{k} y_{Si} \end{aligned} \tag{4.6}$$

4.3.4 The search area

Searching in the whole frame would be time consuming. Two frames are separated by a small lapse of time. For example, let us consider a video with 25 fps. It means that two frames are separated by only $40ms$. Therefore, we can realistically admit that the eye region in the next frame will be in a window the size of which is the double of the found eye region size. Theoretically, given w_e and h_e respectively the width and the height of the regions containing eyes, the maximum distance of displacement of this region between two frames is $\frac{1}{2}w_e$ on x-axis and $\frac{1}{2}h_e$ on y-axis. Let us take a face located at the right side of a given frame, its width is 1/10 of the frame width. This face is moving horizontally to the right side in the video with a horizontal displacement of $\frac{1}{2}w_e$ between two frames. This face will be outside the video window after 20 frames or $0.8s$. Similarly, about the face scale, in each new frame, the eye region can have an area 4 times bigger than the one in the previous frame and still be tracked. After a few frames, the face can be so large that the video window can no longer contain it.

Let us call $(x_e(T), y_e(T))$ the coordinates of the left upper corner of the region containing eyes $Re(T)$ at the frame T and $w_e(T)$, $h_e(T)$ respectively the eye region width and height at the frame T. Hence, we define the search area upper left point position (X_S, Y_S), the width $W_S(T)$ and the height $H_S(T)$ at frame T by the equation 4.7.

$$
\begin{aligned}
X_S(T) &= x_e(T-1) - \frac{1}{2}w_e(T-1) \\
X_S(T) &= y_e(T-1) - \frac{1}{2}h_e(T-1) \\
W_S(T) &= 2 \cdot w_e(T-1) \\
H_S(T) &= 2 \cdot h_e(T-1)
\end{aligned}
\tag{4.7}
$$

Obviously, according to our needs, the search area size can be changed. However, as the search area size is high, the computation time for localizing the eyes will be high too.

4.3.5 Selection of the best samples for the purpose of face recognition

At this point, we have the position (x_E, y_E) and size of the region containing both eyes, we will explain in this section how to extract some of the best samples. For this purpose, first, we estimate the position and the size of the whole face window from the position and size of the bounding box of the region containing both eyes. Then, to select n samples (n is fixed) of a face in a video sequence, we have to answer to two main questions:

- What are the criteria of the quality of the sample?
- With which temporality, do we have to select the samples?

4.3.5.1 Face window position and size estimation

Once we have computed the position and the size of an eye region in a frame, we can approximately estimate the whole face position and size. We do not need an accurate estimation of the face window. We only need a face window in the same order of magnitude as those detected, for example Viola and Jones face detector. It must contain all the anatomic regions as main elements of the window.

To estimate the position and size of the face window, we have to estimate the position (X_v, Y_v) of its left upper corner as well as its size L, since the face window is a square. At this point, we have estimated the position (x_E, y_E) and size (w_E, h_E) of the eye region bounding box. The most stable side of the bounding box is the upper side, because this line is composed of the upper contours of each line, the eyebrows, as well as the eyebrow

arch. The lower side is more difficult to detect. Indeed, because of the eyebrow arch, the lower side of eye regions is in shadow. Hence, since the upper side of the region containing both eyes should be the most stable, we assume that estimation of the face window position and size requires only the (x_E, y_E) and the width x_E.

In order to keep the order of magnitude of all samples, we propose that the eye region should have a width w_E of $\frac{2}{3}L$. The vertical symmetry axis of the eye region bounding box is the same as the symmetry axis of the face window bounding box. Hence, the distance between the left or right side between face window and eye region bounding box is $\frac{1}{6}w_E$. Moreover, the forehead has approximately a height of $\frac{1}{4}L$. With this information on face element distribution, we are able to define the position and size of face window by the equation 4.8.

$$\begin{aligned} x_v &= x_E - \frac{w_E}{4} \\ y_v &= y_E - \frac{3}{8}w_E \\ L &= \frac{3}{2}w_E \end{aligned} \tag{4.8}$$

Figure 4.3 shows an example of the estimated position and size of the face window from the position and size of the region containing the eyes. The obtained square window seems quite correct in order to process the face.

Now we have evaluated the position and size of the whole face window, we can select the best samples, but in order to select or not a sample, we must define some quality criteria. We have distinguished two types of criteria, those which are linked to the face pose and those linked to temporal distribution of the selected samples.

4.3.5.2 Face sample quality criteria

As a reminder, the pose is used to select the best samples of a face. Since most of the faces in videos have an approximate null roll value, the pose quality criteria, in the purpose of recognition, are here, null yaw and pitch values. However, a sample can have an appropriate yaw with an important pitch value and vice versa. Hence, it is actually a multi-criteria problem where some configurations can be non comparable. However, we assume that face samples with an appropriate pitch and a non frontal yaw should

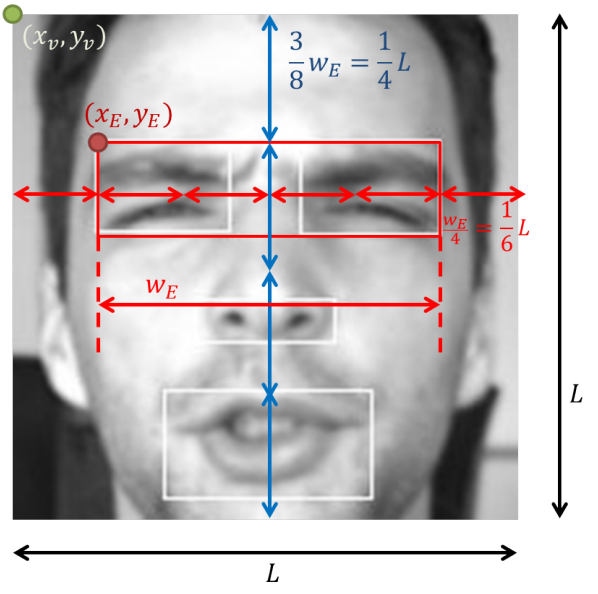

Figure 4.3: Approximate face proportion.

be better for the purpose of face recognition. Thus, we define different rules to order the face images:

1. Samples with which we are able to estimate both yaw and pitch are privileged. Indeed, the table 3.1 of section 2.4.4.3 gives us the rules to estimate the pose depending on the functions E and N associated respectively with spatial distribution of the eyes and nose-mouth spatial distribution. Some configurations can be paradoxical and unrealistic. In such cases, it is impossible to estimate the yaw. Moreover, in order to compute the pitch, we need to detect at least one eye, the nose and the mouth, but one of them can be occluded. Hence, it can happen that the estimation of the yaw or the pitch is not determined.

2. When pitch and yaw are both estimated, a sample with a better pitch (more frontal) is privileged whatever the yaw is. As a consequence, when two samples have almost the same pitch, the one with the most frontal yaw is privileged.

3. Then come the samples where the pitch is estimated, but the yaw is not determined. Indeed, even if the function E has always a value (the starting point of this section is that we have detected an eye region and estimated the whole face window position and size), N can be non determined, for example if the nose is missing. Moreover,

as we said, even if N is determined, the combination of E and N can be paradoxical and hence gives a non determined result.

4. The last case is the worse configuration for the purpose of face recognition. Here, a sample is detected but, we are able to compute neither the function E, nor the function N. Obviously, it is not possible to estimate the pose. It happens when tracking fails or there is an occlusion of the whole face.

The consequence of these rules is that any sample quality can be compared to another sample quality. Hence, all the combinations between the yaw and pitch as described in section 2.4.4.3 can be associated with a discrete and ordered value. As a reminder, the yaw is estimated by 9 discrete values. Since the aim is to have a frontal view, each of the four left side discrete yaws has an equivalent discrete yaw among the four right side yaws. This is symmetric and hence only 5 partial and ordered quality values are sufficient to describe the yaw. However, the yaw can be non determined and hence, 6 ordered values are needed to describe the yaw. Similarly, since the pitch is estimated by 3 discrete poses. Only 3 ordered values are necessary to describe the pitch. With the rules described above, we are able to combine the yaw and the pitch and give an ordered finite number of quality representations, each of them associated with a combination of the possible yaw and pitch. Hence, given V, a face sample, its associated quality measure is defined as $f_q(V)$. The highest the quality is, the highest $f_q(V)$ is.

4.3.5.3 Temporal criterion

in order to prevent the selection of consecutive frames, we have to define some temporal selection criterion.

Let us call $F = \{F_1, F_2, F_3, \cdots, F_n\}$, an ordered set of n samples at different frame indexes. Samples of F are ordered according to an ascending order of the frame indexes. At frame T, another sample F_{n+1} is introduced. We call $F' = \{F_1, F_2, F_3, \cdots, F_n, F_{n+1}\}$, the ordered set defined by n previous samples plus the new sample.

Besides, for each sample, given the temporal distance $d(T_i, T_j)$ which separates the samples T_i and T_j, we want to remove the sample which is the closest (in terms of frames number) to one of its adjacent selected samples. Therefore, we will privilege new selected samples, while removing the element of F' which is the closest to another sample of F'.

For each sample F_i of F', we define a temporal loneliness λ_i which is the minimum between $d(F_i, F_{i-1})$ and $d(F_i, F_{i+1})$ (equation 4.9).

$$\lambda_i = min\left(d(F_i, F_{i-1}), F_i, F_{i+1}\right) \tag{4.9}$$

Here, we will remove the sample F_i of F' which minimizes the temporal loneliness λ_i and then we update $F \leftarrow F'$. The new sample F_{n+1} is included in F, if its loneliness is greater than the loneliness of every sample F_i of F. Hence, we prevent from selecting successive samples.

4.3.5.4 Samples selection: combination of face quality and temporal criteria

In the previous section, we show how the loneliness is computed in a general case and how we prevent the selection of successive samples. Nevertheless, the selected samples are not of equal quality.

S_{T-1} is the set of the best selected samples at frame $T-1$. Here is the summary of the samples selection process:

1. (initialization) The n first samples are always selected in the set containing the best samples S.
2. Let F_T be the new sample, if some samples of S_{T-1} have a lower or same quality measure, one of the samples with the lowest quality measure has to be removed. Indeed, several samples can have the same quality measure. The replacement is processed using the temporal loneliness minimization in $S_{T-1} \cup \{F_T\}$
3. Otherwise, it means that all the samples of S_{T-1} have a greater quality measure than F_T, and hence the process continues with the next frame.

To select the n best samples of a face in a video sequence, let us define more accurately $S_{T-1} = \{S_1, S_2, S_3, \cdots, S_n\}$, the set of n samples selected at frame $T-1$, ordered by a descending order of the quality measure f_q, thus, S_n is the sample with the lowest quality measure in S_{T-1}.

At frame T, the sample F_T pose is estimated giving the associated quality measure $f_q(F_T)$. If $f_q(F_T) < f_q(S_n)$, then the sample F_T is not selected.

Otherwise, if $f_q(F_T) \geq f_q(S_n)$, given the set $S' = \{S_1, S_2, S_3, \cdots, S_n, F_T\}$, we define the subset

$$SQ = \{s/s \in S' \text{ and } f_q(s) = f_q(S_n)\}$$

Then, we remove the sample S_0 which is the sample of the subset SQ which minimizes the temporal loneliness computed on the frame. We build a new set $S_T = S' \backslash \{S_0\}$.

Since we considered both quality and temporal criteria, we should have selected the most frontal view samples and these selected samples should not be temporally successive. The next section will present the evaluation of the tracking method as well as some results of the sample selection method.

4.4 Evaluation

In this section, we will evaluate the tracking method and show some results on the selected samples for the purpose of face recognition.

4.4.1 YouTube Faces database

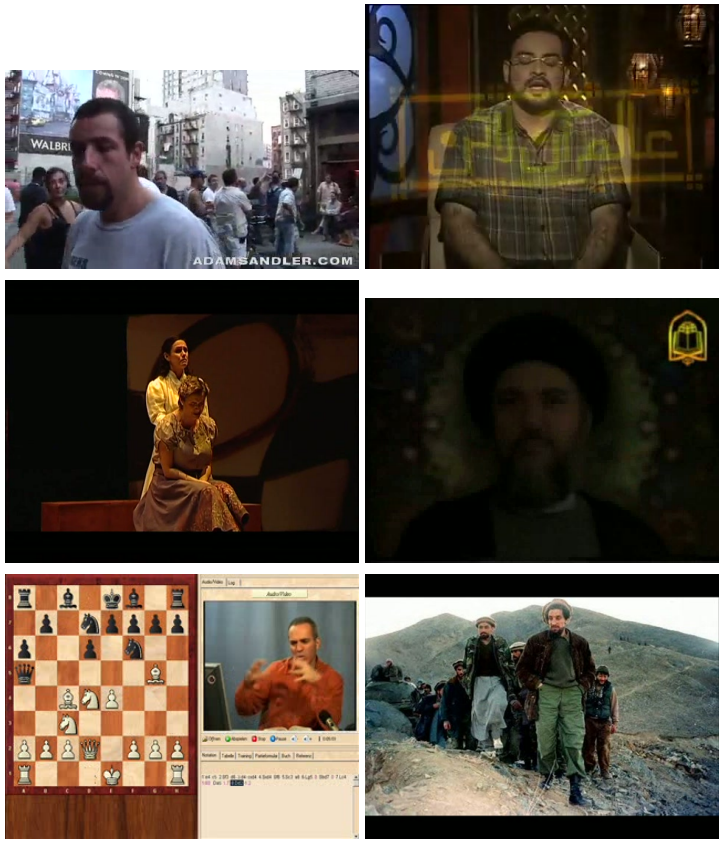

Figure 4.4: Frame examples in YouTube database.

To evaluate our eye tracking method, we use YouTube Faces database described in [Wolf et al., 2011]. The authors inspire the LFW database and created it in 2011. It contains 3425 videos from 1595 different celebrities and videos are downloaded from YouTube. The shortest video sequence duration is 48 frames and the longest one is 6070 frames. The average is 181.3 frames.

Most of the videos have a low resolution and a poor quality. Face scales vary a lot from a video to another. Many videos contain also several faces. Illumination conditions vary a lot and there are also many occlusions on

faces. Figure 4.4 shows some frame examples in YouTube database.

4.4.2 Choosing the similarity function

In our tracking method, we presented two similarity functions. Here, the one used for evaluation is the function defined at connected component level, because with the similarity function at pixel level, the processing time is too long. The processing time to track one eye sample is about $5s$ to $20s$ according to the size of the tracked eye region.

Contrary to the similarity function at pixel level, the one at connected component level enables real time process. Even if we can assume that the process with the function at pixel level should give better results, it cannot be used in a real application.

4.4.3 Tracking results

In order to test, we only take into account videos where Viola and Jones face detector succeeds to detect a face at the first frame of the sequence. All video clips contain at least one face in the first frame, but Viola and Jones face detector finds a face at the frame in only 83.77% of the video clips. Hence, for this evaluation 2869 clips are taken into account. Note that the clips where Viola and Jones face detector fail are not necessarily the most difficult. Viola and Jones face detector can detect small faces, since AdaBoost was performed on face samples of size 16×16. It is quite robust to illumination variation and obviously to scale variation.

In the YouTube Faces database, in 87.12% of the clips, our eye tracker successfully tracks till the end of the clip. In this database, the eye region have been successfully tracked in a sub-clip with 145.2 frames in average. Note the average includes all the clips with less than 145 frames. It shows that the eye tracker is quite efficient.

Our eye tracker fails for three main conditions:

- Tracking fails when the tracked region is very small. Indeed, as the tracked eye region size is low, connected components to track are even smaller (Figure 4.5).

- Tracking also fails when eyes are covered by another object. Indeed, our tracker is a generic tracker. Once we give an area to track, it will continue the tracking. As a result, when another object masks

entirely the eye region, the tracker will track the new object and no longer the eyes (Figure 4.6).

- The tracker also fails when scale of the eyes changes a lot. Indeed, the region we track has a fixed size.

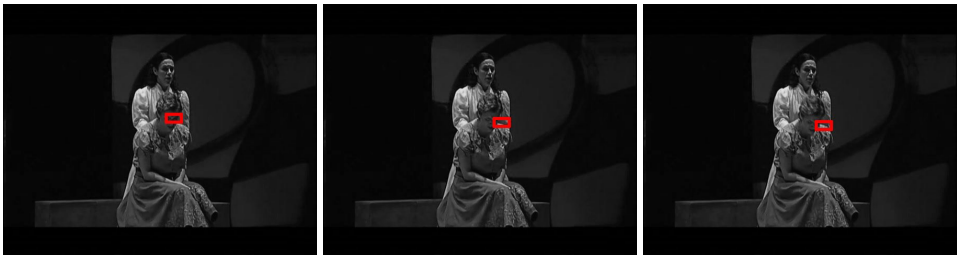

Figure 4.5: Tracking error with very small eye region

Figure 4.6: Tracking error with total occlusion.

However, most of the eye regions in the video clips are tracked with accuracy. Moreover, if a total occlusion of the eye region involves a tracking error. In many clips, small eye regions are well tracked. Besides, our tracker is generic and can be adapted to any object with lines. As we said, sometimes Viola and Jones face detector detects a non face region as a face. It is interesting to see if the tracker can still track the non face object. As we can see in Figure 4.7, our tracker can track other objects than eyes.

Evaluating the efficiency of our best samples is not an easy task. Here, we will only show some results of the best samples selected by our method. Since we assume that frontal faces are the most representative samples of a face in a video clip, we choose video clips where a face is submitted to large pose variation including frontal samples.

Figure 4.8 shows the five best samples of 4 video clips, they are represented in columns. In the first column, four frontal faces are found among 5. In the second column, four frontal faces are also found. In the third column, all faces can be considered as frontal. Finally, in the fourth column, only three samples can be considered as really frontal. However, in any

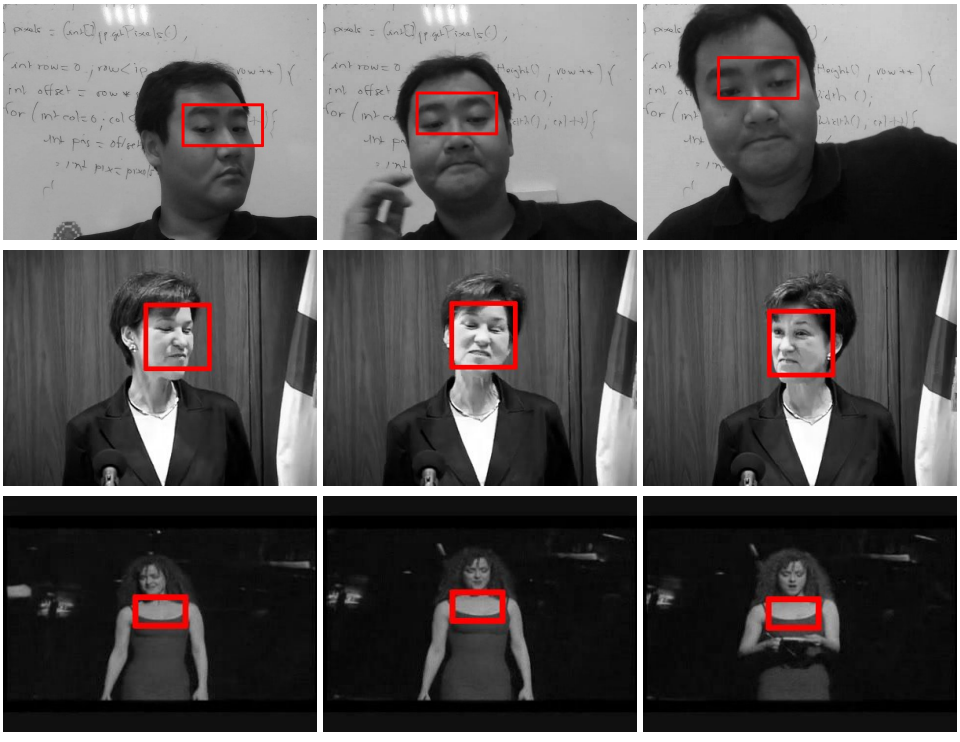

Figure 4.7: Tracking of other objects.

case, at least one sample of frontal face seems to be present in the list of best samples. Moreover, it is important to increase the number of selected best samples, as the number of frames in the video increases too. Globally, evaluating the quality is not obvious at all, we can notice that even the frontal samples are very different because, in particular, of face expression. Our aim was to extract some frontal samples and hence, the whole system achieves this selection of samples in real time.

Figure 4.8: Selection of the five best samples in the video clip.

4.5 Conclusion

In this chapter, we have proposed a method to track an object as we have shown that it is not necessary to track points. Indeed, regions are more efficient to be tracked.

Besides, we have not considered the rough data of pixels in the initial image. We showed it is sufficient to consider a binarized version of the energy map. This enables to have a low computation time. Moreover, the regions we considered could have been modified by the process. So, we have approximated them by simple shapes, rectangles which are the input of the similarity function.

The results we obtain in real time are quite promising and could be included as a element in a large system.

Chapter 5

General conclusion and perspectives

Chapter contents

5.1 General conclusion 152
5.1.1 Back to our face salient region extraction method 152
5.1.2 Back to our head pose estimation methods 153
5.1.3 Back to our tracking method 153
5.1.4 Finding the best face samples 154
5.2 Perspectives 155
5.2.1 Applications of our work 155
5.2.2 Continuation of our work 155
5.2.3 Extension to other objects 156

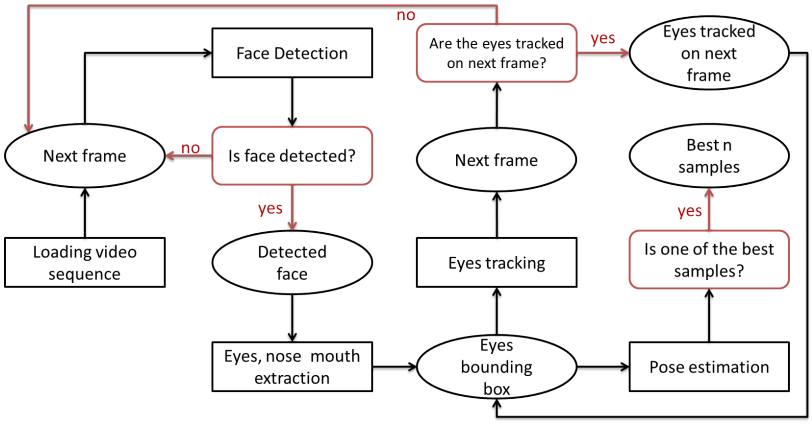

Figure 5.1: Overview of this thesis giving the best face samples among those present in a video sequence.

5.1 General conclusion

The aim of this thesis was to select several face samples among those of a video sequence and on which face recognition algorithms should have better recognition rates. We assume that recognition rates should be higher with frontal faces. Here, we will present the conclusion of the thesis: the benefits of our face salient regions extraction method, of our head pose estimation method and of eye region tracking method.

5.1.1 Back to our face salient region extraction method

First, we wondered which features we can use to reach our goal. We need features robust to all variations (illumination, pose). Obviously, we thought we could use control points approaches such as Active Appearance Models, because they are known to give impressive visual results. However, they have some important limitations. These methods require to localize all the control points, hence, with profile or almost profile view of faces, even if one of the eyes is hidden, all control points, in particular, those of the hidden eye will be localized. Besides, they are known to have some difficulties to generalize: the accuracy of the localization of these control points depends on the learning database.

Moreover, we also show that localizing equivalent control points of the same person in a given video sequence is not an easy task for human vision. Even if equivalent point localization seems visually accurate, we show that, actually, many of these points are localized with a non negligible error. For human vision, a control point of a face is equivalent to many other points (region). Hence, even when visually, control points localization seem impressive, they are not so accurate. Therefore, we choose to extract salient regions instead of control points. The accuracy needed to detect these salient regions should be enough for the purpose of this thesis.

Our face salient region extraction method shows promising results and is quite robust to pose or illumination variations, even with profile views. The extracted regions which are the result of this method can also be used as features of other applications. Notice that the bounding boxes of these salient regions are not the only features which can be used. Indeed, the energy maps, the set of binarized versions of these maps can also be used. Another benefit of our method, contrary to others, is that we can easily understand its mechanism: we, first, thought of the characteristics of the regions we wanted to extract and then chose the filters, contrary to many other methods which use a large amount of features followed by a feature

selection or machine learning technique. Hence, we are able to have an idea of the method efficiency for other problem solving.

5.1.2 Back to our head pose estimation methods

We assume that face salient elements have an approximate horizontal direction. Hence, the local horizontal energy defined previously should have high values with a null roll face. We proved that the global energy of a face is locally a maximum with a null roll face. Thus, we were able to estimate the roll with accuracy using, in particular, this global horizontal face energy and a scoring system based on the credibility of extracted salient regions. Estimating the head roll may be a very important task in many face applications because many methods require an almost null roll face to be efficient.

Moreover, we also present a head yaw and pitch estimation method based on the positions and sizes of the bounding boxes of extracted salient regions. This method relies on the fact that in a given pose, salient region positions and sizes should have a defined configuration regardless the identity of the person. We show that this method gives accurate results with almost null yaw or pitch face values. Since the aim of this thesis is to extract frontal samples of a face in a video sequence, it is possible to use it to find these samples. Another benefit of this method is the time required for processing. Once the salient regions have been extracted, the yaw and the pitch can be estimated with only a few operations. In other words, the time spent to compute the yaw and the pitch is negligible compared to the one needed to extract the face regions.

5.1.3 Back to our tracking method

Our tracking method tracks the region containing both eyes. This region is particularly representative. Tracking only the eye region decreases the computational time. Moreover, if the tracking is accurate enough, we can obtain an approximate face region from this region containing both eyes. As we said, we only need to find a region. When we have the eye region in the current frame, we use the horizontal local energy maps to extract the equivalent region in the next frame. The tracking method is based on the maximization of a similarity measure of these maps. We showed that tracking in the energy maps space is enough to have good results. We also propose two different measures of similarity. The first one is based on computing correlations between binarized versions of horizontal energy

maps. This similarity function gives accurate results, however it is time consuming. The second one can achieve tracking in constant time, it is based on computing the distribution which maximizes the areas of matching connected components in binarized energy maps.

5.1.4 Finding the best face samples

The best face samples are assumed to be frontal ones. Once region containing eyes are tracked, the whole face region is estimated and then the pose can be estimated, in particular, the yaw and the pitch. Instead of extracting only one image per tracked samples, we can choose to extract n samples. Visual results showed that extracted faces are the most frontal. Notice that the most frontal face can be non frontal, when there are no frontal samples in all frames of the video sequence. Hence, the output is not always frontal views.

5.2 Perspectives

In this section, we will present some possible applications of our work and some ideas for the continuation of our work. We will also see if it is possible to use our methods in other objects.

5.2.1 Applications of our work

The proposed methods can be used in many applications. First, it is possible to use them in surveillance systems, for example, with images from security cameras localized at the entry of buildings.

Another application is the determination of the quality of the identity or passport photographs. Indeed, these kinds of images require frontal faces. There are also other criteria of quality for these photographs. For example, faces must have a proper illumination. We will see that one of the idea of the continuation of our work is to determine the illumination quality of a face.

5.2.2 Continuation of our work

We assume that best face samples are those where the roll, the yaw and the pitch are null. However, even if this assumption is true, it is only partial or incomplete. Indeed, other variations can change the appearance of face samples. For example, lightning conditions can vary.

One of the possible continuations of our work is to consider the face illumination conditions among the criteria enabling to select "best faces". Indeed, we can assume that faces with quite homogeneous illumination conditions will lead to higher recognition rates. An idea to estimate the illumination direction is to see how the curve of global horizontal energy according to the roll angle is. As we saw, faces with homogeneous illumination conditions will have only one maximum on this curve. In other faces where light comes from its left or right side will contain some approximate vertical lines, for example on the nose bridge. As the result, several local maximums will appear. However, if we try to find the best lightning conditions as well as the best pose, we will have to deal with a multicriteria optimization problem. Indeed, both are sometimes not compatible: we can have a face with a frontal pose but with bad illumination conditions and vice versa.

Another improvement is to make the tracking adaptive to roll or scale

variations. With the current system, the sizes of all tracked eye regions are the same. However, as the size of this region varies, the size of the tracked eye should also vary. Furthermore, the pose, in particular the yaw and the pitch, is estimated once the eye region is tracked. However, we should introduce to the system a roll estimation before tracking the region containing the eyes. With these improvements, the tracking should be invariant to scale and rotation variations.

Furthermore, if we are able to find the salient regions of human face using the energy maps, we can wonder if it is also possible to detect face in the energy spaces.

5.2.3 Extension to other objects

We can also wonder if it is possible to generalize to other issues some of the methods presented in this thesis. First, the salient region extraction method can be used with objects containing mainly vertical or horizontal lines. We know that the required sizes of Haar patterns must be equivalent to the size of the regions we want to extract. Human eyes, nose and mouth have an approximate horizontal direction and they have a size at the same order of magnitude. For other objects, the regions we want to extract can be at different scales. Hence, for such cases, we will probably need to study several energy maps computed from Haar filters of different sizes. Another requirement to adapt the extraction methods to other objects is the need to build a model of the distribution of horizontal and vertical lines.

Concerning our roll estimation method, we can also probably adapt it to any object with horizontal lines. However, the scoring system must be modified according to properties of the extracted regions. The yaw and pitch estimations will depend on the built model too.

The tracking method as described in this thesis is more general. Tracking any region is actually possible. The only step we have to modify is the initialization step: indeed, we must choose the region to track. Since tracking is also based on the measure of similarity of energy maps, the tracking should be more efficient with objects containing vertical or horizontal lines.

Many objects can be described with the straight lines inside them and thus, our method can be adapted for them. A model of these lines distribution has only to be created. Here, we proposed a methodology which can be applied in many practical applications. For instance, from a camera installed in front of a road, cars running on this road will have mainly approximate horizontal and vertical lines. The methodology we proposed can be adapted to counting cars in order to evaluate the car traffic.

Personal Publications

This is the list of personal publications:

- Pyun, N.-J., Vincent, N., 2015. Head roll estimation using horizontal energy maximization. Intern. Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS), Catania, Italie.
- Pyun, N.-J., Sayah, H., Vincent, N., 2014. Adaptive Haar-Like Features for Head Pose Estimation. Intern. Conf. on Image Analysis and Recognition (ICIAR), Lecture notes in Computer Science 8815, Vilamoura, Portugal, pp94–101.
- Pyun, N.-J., Marmouget, M., Vincent, N., 2014. Détection des yeux, du nez et de la bouche par filtres de Haar adaptatifs. Compression et Représentation des Signaux Audiovisuels (CORESA),Reims, France.
- Pyun, N.-J., Denèfle, S., Vincent, N., 2011. Polygons extraction and graph-matching - detection of mullioned windows. Hypermédias et pratiques numériques (H2PTM), pp367-369.

Bibliography

Ahonen, T., Hadid, A., Pietikäinen, M., 2004. Face Recognition with Local Binary Patterns. ECCV. *Cited page 17*

Akhloufi, M., Bendada, A., 2010. Locally adaptive texture features for multispectral face recognition. Systems, Man and Cybernetics, 3308–3314. *Cited page 11*

Asteriadis, S., Nikolaidis, N., Pitas, I., 2009. Facial feature detection using distance vector fields. Pattern Recognition 42 (7), 1388–1398. *Cited page 74*

Belkin, M., Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396. *Cited page 92*

Beymer, D., 1994. Face recognition under varying pose. Computer Vision and Pattern Recognition, 756–761. *Cited page 87*

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press. *Cited page 90*

Bradski, G. R., 1998. Computer Vision Face Tracking For Use in a Perceptual User Interface. *Cited page 129*

Brubaker, S., Wu, J., et al., J. S., 2008. On the Design of Cascades of Boosted Ensembles for Face Detection. International Journal of Computer Vision 77, 65–86. *Cited page 16*

Charoensak, C., 2004. Face contour tracking in video using active contour model. ICIP 2, 1021–2024. *Cited page 127*

Chu, B., Romdhani, S., Chen, L., 2014. 3D-Aided Face Recognition Robust to Expression and Pose Variations. CVPR, 1907–2014. *Cited page 20*

Chum, ., Matas, J., 2008. Optimal Randomized RANSAC. Pattern Analysis and Machine Intelligence 30, 1472–1482. *Cited page 22*

Comaniciu, D., Ramesh, V., Meer, P., 2000. Real-time tracking of non-rigid objects using mean shift. CVPR 2, 142–149. *Cited page 126*

Cootes, T., Edwards, G., Taylor, C., 2001. Active appearance models. Pattern Analysis and Machine Intelligence 23 (6), 681–685. *Cited page 11*

Cootes, T., Taylor, C., 1999. mixture model for representing shape variation. Image and Vision Computing 17, 567–573. *Cited page 130*

Cootes, T., Walker, K., Taylor, C., 2000. View-based active appearance models. Intern. Conf. on Automatic Face and Gesture Recognition, 227–232. *Cited page 89*

Dahmane, A., Larabi, S., Djeraba, C., 2010. Detection and analysis of symmetrical parts on face for head pose estimation. ICIP, 3249–3252. *Cited page 88*

Dai, J., Chung, R., 2011. Head pose estimation by imperceptible structured light sensing. Intern. Conf. on Robotics and Automation, 1646–1651. *Cited page 89*

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. CVPR 1, 886–893. *Cited page 17*

Del Moral, P., 1995. Markov Processes and Related Fields. Polymat. *Cited page 127*

Destrero, A., Odone, F., Verri, A., 2007. A system for face detection and tracking in unconstrained environments. Advanced Video and Signal Based Surveillance, 499–504. *Cited page 128*

Duda, r., Hart, P., Stork, D., 2001. Pattern Classification. John Wiley and Sons. *Cited page 91*

Foyti, J., Sankaran, P., Asari, V., 2011. RTracking and Recognizing Multiple Faces Using Kalman Filter and ModularPCA. Procedia Computer Science 6, 256–261. *Cited page 126*

Gee, A., Cipolla, R., 1994. Determining the gaze of faces in images. Image and Vision Compution, 639–647. *Cited page 88*

Gizatdinova, Y., Surakka, V., 2007. Automatic Detection of Facial Landmarks from AU-coded Expressive Facial Images. ICIAP, 419–424. *Cited page 18*

Grauman, K., Darrell, T., 2004. Fast contour matching using approximate earth mover's distance. CVPR 1, 220–227. *Cited page 130*

Haj, M., Gonzalez, J., Davis, L., 2012. On partial least squares in head pose estimation: How to simultaneously deal with misalignment. CVPR, 2602–2609. *Cited page 90*

Hartley, R., Sturm, P., 1995. Triangulation. Computer Analysis of Images and Patterns 970, 190–197. *Cited page 22*

Hoffken, M., Wang, T., Wiest, J., Kressel, U., Dietmayer, K., 2013. Synchronized Submanifold Embedding for Robust and Real-Time Capable Head Pose Detection Based on Range Images. Intern. Conf. on 3D Vision, 167–174. *Cited page 91*

Hu, Y., Chen, L., Zhou, Y., Zhang, H., 2004. Estimating face pose by facial asymmetry and geometry. Intern. Conf. on Automatic Face and Gesture Recognition, 154–156. *Cited page 92*

Huang, F.-Z., Su, J.-B., 2004. Face contour detection and tracking with complex backgrounds. Intern. Conf. on Machine Learning and Cybernetics 6, 3855–3859. *Cited page 127*

Huang, J., Shao, X., Wechsler, H., 1999. Face pose discrimination using support vector machines (SVM). Intern. Conf. on Patter. Recog., 154–156. *Cited page 87*

Huang, K., Trivedi, M., 2003. Video arrays for real-time tracking of person, head, and face in an intelligent room. Machine Vision and Applications, 103–111. *Cited page 92*

Hui, T., Yi-qin, C., Ting-zhi, S., 2010. Face tracking using multiple facial features based on particle filter. Intern. Asia Conf. on Informatics in Control, Automation and Robotics 3, 72–75. *Cited page 127*

Isard, M., Blake, A., 1998. CONDENSATION—Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision 29, 5–28. *Cited page 127*

Jain, A., Unsang, P., 2009. Facial marks: Soft biometric for face recognition. ICIP, 37–40. *Cited page 11*

Jesorsky, O., Kirchberg, K., Frisholz, R., 2001. Robust face detection using the Hausdorff distance. In: Audio and video based Person Authentication. LNCS, 90–95. *Cited page 74*

Jian, W., Honglian, Z., 2009. Eye detection based on mult-angle template matching. Image Analysis and Signal Processing, 241–244. *Cited page 18*

Jiang, M., Deng, L., Zhang, L., Tang, J., Fan, C., 2012. Head pose estimation based on Active Shape Model and Relevant Vector Machine. Intern. Conf. on Systems, Man, and Cybernetics, 1035–1038. *Cited page 89*

Jin, H., Liu, Q., Lu, H., 2004. Face detection using improved LBP under bayesian framework. International Conference on Image and Graphics, 306–309. *Cited page 17*

Jones, M., Viola, P., 2003. Fast Multi-view Face Detection. Computer Vision and Pattern Recognition. *Cited pages 16 and 87*

Juefei-Xu, F., Luu, K., Savvides, M., Bui, T., Suen, C., 2011. Investigating age invariant face recognition based on periocular biometrics. Intern. Joint Conf. on Biometrics Compendium, 1–7. *Cited page 131*

Kailath, T., 1967. The Divergence and Bhattacharyya Distance Measures in Signal Selection. Trans. on Communication Technology 15, 52–60. *Cited page 126*

Kawato, S., Ohya, J., 2000. Automatic skin-color distribution extraction for face detection and tracking. Intern. Conf. on Signal Processing 2, 1415–1418. *Cited page 128*

Kim, J.-H., Kang, B.-D., Eom, J.-S., Kim, C.-S., Ahn, S.-H., Shin, B.-J., Sang-Kyoon, K., 2007. Real-Time Face Tracking System Using Adaptive Face Detector and Kalman Filter. HCI Intelligent Multimodal Interaction Environments, Lecture Notes in Computer Science 4552, 669–678. *Cited page 126*

Kobayashi, A., Satake, J., Hirayama, T., Kawashima, H., Matsuyama, T., 2008. Person-independent face tracking based on dynamic AAM selection. Intern. Conf. on Automatic Face and Gesture Recognition, 1–8. *Cited page 129*

Kong, S., Mbouna, R., 2015. Head Pose Estimation From a 2D Face Image Using 3D Face Morphing With Depth Parameters. Image Processing, 1801–1808. *Cited page 88*

Kotropoulos, C., Pitas, I., 1997. Rule-based face detection in frontal views. Acoustics, Speech and Signal Processing 4, 2537–2540. *Cited page 18*

Krinidis, M., Nikolaidis, N., Pitas, I., 2007. 2-D Feature-Point Selection and Tracking Using 3-D Physics-Based Deformable Surfaces. Trans. on Circuits and Systems for Video Technology 17, 876–888. *Cited page 130*

Kruger, V., Happe, A., Sommer, G., 2000. Affine real-time face tracking using Gabor wavelet networks. Intern. Conf. on Pattern Recognition, 127–130. *Cited page 128*

Krüger, V., Sommer, G., 2002. Gabor wavelet networks for efficient head pose estimation. Image and Vision Computing 20, 665–672. *Cited page 90*

Lanitis, A., Taylor, C., Cootes, T., Ahmed, T., 1995. Automatic Interpretation of Human Faces and Hand Gestures Using Flexible Models. Intern. Conf. on Automatic Face and Gesture Recognition, 98–103. *Cited page 89*

Lefèvre, S., Vincent, N., 2004. Real Time Multiple Object Tracking Based on Active Contours. In: Intern. Conf. on Image Analysis and Recognition. pp. 606–613. *Cited page 127*

Levi, K., Weiss, Y., 2004. Learning object detection from a small number of examples: the importance of good features. CVPR 2, 53–60. *Cited page 17*

Li, S. Z., Zhu, L., Zhang, Z., 2002. Statistical Learning of Multi-view Face Detection. ECCV 2353, 67–81. *Cited page 15*

Li, Y., fei Zhao, P., kun Wan, B., Ming, D., 2008. An Improved Hybrid Projection Function for Eye Precision Location. MIMI 4987, 312–321. *Cited page 74*

Li, Y., Gong, S., Liddell, H., 2001. Modelling faces dynamically across views and over time. Intern. Conf. on Computer Vision 1, 554–559. *Cited page 130*

Li, Y., Gong, S., Liddell, H., 2004. Support vector machine based multi-view face detection and recognition. Image and Vision Computing, 413–427. *Cited page 90*

Lienhart, R., Maydt, J., 2002. An extended set of Haar-like features for rapid object detection. ICIP 1, 900–903. *Cited page 15*

Luu, K., Seshadri, K., Savvides, M., Bui, T., Suen, C., 2011. Contourlet appearance model for facial age estimation. Intern. Joint Conf. on Biometrics Compendium, 1–8. *Cited page 90*

Marquardt, D., 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journ. of the Society for Industrial and Applied Mathematics 11, 431–441. *Cited page 128*

Martins, P., Batista, J., 2008. Single view head pose estimation. ICIP, 1652–1655. *Cited page 89*

McKenna, S., Gong, S., 1996. Tracking Faces. Intern. Conf. on Automatic Face and Gesture Recognition, 271–276. *Cited page 126*

Meynet, J., Popovici, V., Thiran, J.-P., 2007. Face detection with boosted Gaussian features. Pattern Recognition 40, 2283–2291. *Cited page 16*

Mikami, D., Otsuka, K., Yamato, J., 2009. Memory-based Particle Filter for face pose tracking robust under complex dynamics. CVPR, 999–1006. *Cited page 127*

Murphy-Chutorian, E., Trivedi, M., 2009. Head Pose Estimation in Computer Vision: A Survey. Pattern Analysis and Machine Intelligence 31 (14), 607–626. *Cited page 11*

Murphy-Chutorian, E., Trivedi, M., 2010. Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness. Intelligent Transportation Systems 11, 300–311. *Cited page 90*

Niu, D.-J., Zhan, Y., Song, S.-L., 2003. Research and implementation of real-time face detection, tracking and protection. Intern. Conf. on Machine Learning and Cybernetics 5, 2765–2770. *Cited page 128*

Niyogi, S., Freeman, W., 1996. Example-based head tracking. Intern. Conf. on Automatic Face and Gesture Recognition, 374–378. *Cited page 87*

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence 24, 971–987. *Cited page 16*

Opelt, A., Pinz, A., Zisserman, A., 2006. A Boundary-Fragment-Model for Object Detection. ECCV 3952, 575–588. *Cited page 17*

Osadchy, R., Miller, M., LeCun, Y., 2007. Synergistic face detection and pose estimation with energy-based model. Machine Learning Research 8, 1197–1215. *Cited page 90*

Park, M., Lee, S., 2008. Face modeling and tracking with Gabor Wavelet Network prior. Intern. Conf. on Pattern Recognition, 1–4. *Cited page 128*

Pyun, N.-J., Marmouget, M., Vincent, N., 2014a. Dt́ection des yeux, du nez et de la bouche par filtres de Haar adaptatifs. Compression et Représentation des Signaux Audiovisuels (CORESA). *Cited page 49*

Pyun, N.-J., Sayah, H., Vincent, N., 2014b. Adaptive Haar-Like Features for Head Pose Estimation. Intern. Conf. on Image Analysis and Recognition, Lecture notes in Computer Science 8815, 94–101. *Cited page 103*

Pyun, N.-J., Vincent, N., 2015. Head roll estimation using horizontal energy maximization. Advanced Concepts for Intelligent Vision SystemsThis paper will be persented in october 2015. *Cited page 94*

Rae, R., Ritter, H., 1998. Recognition of human head orientation based on artificial neural networks. Trans. on Neural Networks 9, 257–265. *Cited page 90*

Raytchev, B., Yoda, I., Sakaue, K., 2004. Head pose estimation by nonlinear manifold learning. Intern. Conf. on Pattern Recognition 4, 462–466. *Cited page 92*

Roweis, S., Saul, L., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326. *Cited page 92*

Sabzmeydani, P., Mori, G., 2007. Detecting Pedestrians by Learning Shapelet Features. CVPR 1, 1–8. *Cited page 17*

Seshadri, K., Savvides, M., 2009. Robust modified Active Shape Model for automatic facial landmark annotation of frontal faces. Intern. Conf. on Biometrics: Theory, Applications, and Systems, 1–8. *Cited page 90*

Sherrah, J., Gong, S., Ong, E.-J., 2001. Face distributions in similarity space under varyinghead pose. Image and Vision Computing, 807–819. *Cited page 91*

Sozou, P., Cootes, T., Taylor, C., Di Mauro, E., 1995. Nonlinear generalization of point distribution models using polynomial regression. Image and Vision Computing 13, 451–457. *Cited page 130*

Srinivasan, S., Boyer, K., 2002. Head pose estimation using view based eigenspaces. Intern. Conf. on Pattern Recognition, 302–305. *Cited page 91*

Stamou, G., Nikolaidis, N., Pitas, I., 2005. Object tracking based on morphological elastic graph matching. ICIP, 709–712. *Cited page 129*

Stasiak, L., Pacut, A., 2007. Particle filters for multi-face detection and tracking with automatic clustering. Imaging Systems and Techniques workshop, 1–6. *Cited page 127*

Su, Y., Ai, H., Lao, S., 2008. Real-time face alignment with tracking in video. ICIP, 1632–1635. *Cited page 129*

Su, Y., Ai, H., Lao, S., 2009. Multi-View Face Alignment Using 3D Shape Model for View Estimation. Computer Science 5558, 179–188. *Cited page 130*

Tan, X., Song, F., Zhou, Z.-H., Chen, S., 2009. Enhanced Pictorial Structures for precise eye localization under incontrolled conditions. CVPR, 1621–1628. *Cited pages 75 and 77*

Tan, X., Triggs, A., 2010. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. Biometrics Compendium 19 (6), 1635–1650. *Cited page 11*

ter Braak, C., Vrugt, J., 2008. Differential Evolution Markov Chain with snooker updater and fewer chains. Statistics and Computing 18, 435–446. *Cited page 127*

Vadakkepat, P., Lim, P., De Silva, L., Jing, L., Ling, L. L., 2008. Multimodal Approach to Human-Face Detection and Tracking. Industrial Electronics 65, 1385–1393. *Cited page 128*

Voit, M., Nickel, K., Stiefelhagen, R., 2007. Head pose estimation in single- and multi-view environments results on the CLEAR'07 benchmarks. CLEAR. *Cited page 90*

Vukainovic, D., Pantic, M., 2005. "Fully automatic facial feature point detection using Gabor feature based boosted classifiers. Systems, Man and Cybernetics 2, 1692–1698. *Cited page 11*

Wang, J.-G., Sung, E., 2007. EM enhancement of 3D head pose estimated by point at infinity. Image and Vision Computing, 1864–1874. *Cited page 88*

Wang, P., Ji, Q., 2005. Learning discriminant features for multi-view face and eye detection. CVPR 1, 373–379. *Cited page 16*

Wang, X., Han, T. X., Yan, S., 2009. An HOG-LBP human detector with partial occlusion handling. ICCV, 32–39. *Cited page 17*

Wilson, H., Wilkinson, F., Lin, L., Castillo, M., 2000. Perception of head orientation. Vision Research, 459–472. *Cited page 88*

Wolf, L., Hassner, T., Maoz, I., 2011. Face recognition in unconstrained videos with matched background similarity. CVPR, 529–534. *Cited page 144*

Wu, B., Ai, H., Huang, C., 2004. Fast rotation invariant multi-view face detection based on real AdaBoost. In: In Sixth IEEE International Conference on Automatic Face and Gesture Recognition. pp. 79–84. *Cited page 16*

Wu, B., Nevatia, R., 2005. Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. ICIP 1, 90–97. *Cited page 17*

Wu, J., Trivedi, M., 2008. A two-stage head pose estimation framework and evaluation. Intern. Conf. on Pattern Recognition, 1138–1158. *Cited pages 91 and 92*

Xia, S., Li, J., Xia, L., 2006. Robust Face Tracking Using Self-Skin Color Segmentation. Intern. Conf. on Signal Processing 2. *Cited page 129*

Xiangrong Chen, L. G., Li, S., Zhang, H.-J., 2001. Learning representative local features for face detection. CVPR 1, 1126–1131. *Cited page 16*

Xiao, J., xiang Chai, J., Kanade, T., 2004. A Closed-Form Solution to Non-rigid Shape and Motion Recovery. Computer Vision 3024, 573–587. *Cited page 130*

Yan, S., Shan, S., Chen, X., 2008a. Locally Assembled Binary (LAB) feature with feature-centric cascade for fast and accurate face detection. CVPR, 1–7. *Cited page 17*

Yan, S., Zhang, Z., Fu, Y., Hu, Y., Tu, J., Huang, T., 2008b. Learning a person-independent representation for precise 3d pose estimation. Multimodal Technologies for Perception of Humans, Lecture Notes in Computer Science 4625, 297–306. *Cited page 92*

Yang, J., Liang, W., Jia, Y., 2012. Face pose estimation with combined 2D and 3D HOG features. ICPR, 2492–2495. *Cited page 90*

Yuille, A., Cohen, D., Hallinan, P., 1989. Feature extraction from faces using deformable templates. CVPR, 104–109. *Cited page 18*

Yun, T., Guan, L., 2010. Fiducial point tracking for facial expression using multiple particle filters with kernel correlation analysis. ICIP, 373–376. *Cited page 127*

Zhang, L., Chu, R., Xiang, S., 2007a. Face detection based on multi-block LBP representation. international conference on Advances in Biometrics, 11–18. *Cited page 17*

Zhang, Z., Hu, Y., Liu, M., Huang, T., 2007b. Head pose estimation in seminar room using multi view face detectors. Workshop Classification of Events Activities and Relationships, ser. Lecture Notes in Computer Science, 299–304. *Cited page 87*

Zhao, S., Precioso, F., Cord, M., 2009. Spatio-Temporal Tube Kernel for Actor Retrieval. ICIP, 1885–1888. *Cited page 130*

Zhong, L., Liu, Q., Yang, P., Metaxas, M., 2012. Leanrning active facial patches for expression analysis. CVPR, 2562–2569. *Cited page 11*

Zhou, M., Liang, L., Sun, J., Wang, Y., 2010a. AAM based face tracking with temporal matching and face segmentation. CVPR, 701–708. *Cited page 11*

Zhou, M., Liang, L., Sun, J., Wang, Y., 2010b. AAM based face tracking with temporal matching and face segmentation. CVPR, 701–708. *Cited page 129*

Zhou, M., Wang, Y., Huang, X., 2010c. Real-Time 3D Face and Facial Action Tracking Using Extended 2D+3D AAMs. ICPR, 3963–3966. *Cited page 129*

Zhu, Y., Fujimura, K., 2004. Head pose estimation for driver monitoring. Intelligent Vehicles Symposium, 501–506. *Cited page 92*

Zhu, Z., Ji, Q., 2004. 3D face pose tracking from an uncalibrated monocular camera. ICPR 4, 400–403. *Cited page 126*