HAL open science

Analyse faciale avec dérivées Gaussiennes

John Alexander Ruiz Hernandez

To cite this version:

John Alexander Ruiz Hernandez. Analyse faciale avec dérivées Gaussiennes. Mathématiques générales [math.GM]. Université de Grenoble, 2011. Français. NNT : 2011GRENM039 . tel-00646718

HAL Id: tel-00646718

https://theses.hal.science/tel-00646718v1

Submitted on 30 Nov 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique et Mathématique Appliquée**

Arrêté ministérial : 7 aôut 2006

Présentée par

John Alexander RUIZ HERNANDEZ

Thèse dirigée par **James L. CROWLEY**
et codirigée par **Augustin LUX**

préparée au sein **du Laboratoire d'Informatique de Grenoble á l'INRIA Rhône-Alpes**
et de **l'Ecole Doctorale de Mathématiques, Sciences et Technologies de l'Information**

Facial Analysis with Gaussian Derivatives

Thèse soutenue publiquement le **23 Septembre 2011**,
devant le jury composé de :

M. Peter STURM

Research Director at INRIA Rhône-Alpes, Président

M. Bruce DRAPER

Professor at Colorado State University, Rapporteur

M. Frederic JURIE

Professor at The Universite of Caen, Rapporteur

M. James M. REHG

Professor at Georgia Institute of Technology, Examinateur

M. James L. CROWLEY

Professor at INPG -ENSIMAG, Directeur de thèse

M. Augustin LUX

Professor at INPG -ENSIMAG, Co-Directeur de thèse

To my family, specially to my parents Jose and Olga, without whom it would not have been possible. This is also for you, my two beloved brothers in heaven..

A mi familia, especialmente a mis padres, Jose y Olga, sin ellos este sueño no hubiese sido posible. Esto también va por ustedes mis dos amados hermanos en el cielo..

Acknowledgments

I WOULD like to thank all those people who have contributed in different ways to the work that is presented in this thesis.

First of all, I would like to thank my thesis advisors and co-workers, James L. Crowley and Augustin Lux, for their motivation, their guidance and numerous discussions. I am also grateful with Peter Sturm, James Rehg, Bruce Draper and Frederic Jurie for the interest in my work and for being members of the thesis jury.

For friendship and research collaboration, thanks to Nicolas Gourier, Claudine Combe and Antoine Meler for the fruitful discussions, smiles, travels, coding advice and for sharing an office in this tough period of my life. Thank you to Carlos Jaimes Barrios and Ana Amaya for their collaboration and support during this academic endeavour. Thank you to Evanthia Mavridou and Judith Brenner who have patiently corrected the thesis. Thank you to Charlotte Beaussier for her energy, happiness, craziness and friendship whose filled my head and my heart during all this time in french lands and in the last harder times of the thesis redaction.

Thank you to all the current and former members of the Prima Group, Remi B., Jean-Pascal, Sofia, Remi E., Thibault, Silvain, Harsimrat, Patrick, Dominique, Alexandre, Frederic, Amaury ,Matthieu L., Mathieu G-B. and Mathieu V. for the nice atmosphere in the group. It was a pleasure to work to you in Grenoble. Thank you guys, I've had the time of my life with all of you.

Finally, there is no way that I would have been able to make it this far without the unbounded love and support of my wonderful family, who taught me everything. I really needed to know anyway. Moms, Dads, Nephews, nieces Fabio, Eloisa, Carlos and Liliana, I owe it all to you.

John Alexander RUIZ HERNANDEZ
Grenoble, November 30, 2011

Facial Analysis with Gaussian Derivatives

Abstract:

In this thesis, we explore the use of multi-scale Gaussian Derivatives as an initial representation for detection, recognition and classification of human faces in images. We show that a fast, $O(N)$, binomial pyramid algorithm can be used to provide Gaussian derivatives with identical sampled impulse responses at scale factors of $\sqrt{2}$. We then show that a vector of such derivatives at multiple scales and derivative orders for each pixel can be used as basis for algorithms for detection, classification and recognition that meet or exceed state of the art performance with reduced computation cost. Furthermore, the use of integer coefficients and $O(N)$ complexity in computation and memory requirements make such an approach suitable for real time applications running in embedded image processing on mobile devices.

We test this representation using three classic problems of facial image analysis: Face detection, face recognition and age estimation. For face detection, we investigate multi-scale Gaussian derivatives as an alternative to Haar wavelets for use with a cascade of linear classifiers learned with the Adaboost algorithm, as made popular by Viola and Jones. We show that the pyramid representation can be use to optimize the detection process by adapting the position of derivatives in the cascade. In these experiments we are able to show that we can obtain similar detection performance levels (as measured by ROC curves) with an important reduction in computation cost. For face recognition and age estimation, we show that multi-scale Gaussian derivatives can be used to compute a tensorial representation that retains the most important facial information. We show that when combined with Multilinear Principal Component Analysis and Kernel Discriminative Common Vectors (KDCV) can lead to an algorithm that are similar to competing techniques for face recognition at reduced computational cost. For age estimation from facial images, we show that our tensorial representation using multi-scale Gaussian derivatives can be used with a relevance vector machine to provide age estimation at performance levels that are similar to state of the art methods.

Keywords: Gaussian derivatives, Face detection and Recognition, Age estimation, Half-Octave Gaussian Pyramid, Facial analysis

Analyse Faciale avec les Dérivés Gaussiennes

Résumé: Dans cette thèse, nous explorons l'utilisation des dérivées Gaussiennes multi-échelles comme représentation initiale pour la détection, la reconnaissance et la classification des visages humains dans des images. Nous montrons qu'un algorithme rapide, $O(N)$, de construction d'une pyramide binomiale peut être utilisé pour extraire des dérivées Gaussiennes avec une réponse impulsionnelle identique à un facteur d'échelle $\sqrt{2}$. Nous montrons ensuite qu'un vecteur composé de ces dérivées à différentes échelles et à différents ordres en chaque pixel peut être utilisé comme base pour les algorithmes de détection, de classification et de reconnaissance lesquels atteignent ou dépassent les performances de l'état de l'art avec un coût de calcul réduit. De plus l'utilisation de coefficients entiers, avec une complexité de calcul et des exigences mémoires en $O(N)$ font qu'une telle approche est appropriée pour des applications temps réel embarquées sur des systèmes mobiles.

Nous testons cette représentation en utilisant trois problèmes classiques d'analyse d'images faciales : détection de visages, reconnaissance de visages et estimation de l'âge. Pour la détection de visages, nous examinons les dérivées Gaussiennes multi-échelles comme une alternative aux ondelettes de Haar pour une utilisation dans la construction d'une cascade de classifieurs linéaires appris avec l'algorithme Adaboost, popularisé par Viola and Jones. Nous montrons que la représentation pyramidale peut être utilisée pour optimiser le processus de détection en adaptant la position des dérivées dans la cascade. Dans ces experiences nous sommes capables de montrer que nous pouvons obtenir des niveaux de performances de détection similaires (mesurés par des courbes ROC) avec une réduction importante du coût de calcul. Pour la reconnaissance de visages et l'estimation de l'âge, nous montrons que les dérivées Gaussiennes multi-échelles peuvent être utilisées pour calculer une représentation tensorielle qui conserve l'information faciale la plus importante. Nous montrons que combinée à l'Analyse Multilinéaire en Composantes Principales et à la méthode Kernel Discriminative Common Vectors (KDCV), cette représentation tensorielle peut mener à un algorithme qui est similaire aux techniques concurrentes pour la reconnaissance de visages avec un coût de calcul réduit. Pour l'estimation de l'âge à partir d'images faciales, nous montrons que notre représentation tensorielle utilisant les dérivées de Gaussiennes multi-échelles peut être utilisée avec une machine à vecteur de pertinence pour fournir une estimation de l'âge avec des niveaux de performances similaires aux méthodes de l'état de l'art.

Mots-Clés: Gaussian derivatives, Face detection and Recognition, Age estimation

John RUIZ-HERNANDEZ's Publications related with this thesis

1. **J.A. Ruiz Hernandez**, A. Lux and J.L. Crowley, "Face Detection by Cascade of Gaussian Derivates Calculated with a Half-Octave Gaussian Pyramid," in *Proc. of IEEE conference on Automatic Face and Gesture Recognition*, Amsterdam (Netherlands), Sep. 2008.

2. **J.A. Ruiz Hernandez**, J.L. Crowley, A. Meler and A. Lux, "Face Recognition Using Tensors Of Census Transform Histograms From Gaussian Features Maps," in *Proc. of British Machine Vision Conference*, London (UK), Sep. 2009.

3. **J.A. Ruiz Hernandez**, J.L. Crowley and A. Lux, "Tensor-Jet: A Tensorial Representation of Local Binary Gaussian Jet Maps," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, San Francisco (USA), Jun. 2010.

4. **J.A. Ruiz Hernandez**, J.L. Crowley and A. Lux, "How old are you?" : Age Estimation with Tensors of Binary Gaussian Receptive Maps," in *Proc. of British Machine Vision Conference*, Aberystwyth(UK), Sep. 2010.

List of Supplemental Publications

The following publications by the author are also related to the thesis subjects, but are not included as a part of the thesis.

1. A. Meler, **J.A. Ruiz Hernandez** and J.L. Crowley, "Probabilistic Model of Error in Fixed-Point Arithmetic Gaussian Pyramid," in *Proc. of IEEE 12th International Conference on Computer Vision Workshops*, Kyoto (Japan) Sep. 2009, pp. 816–820.

Contents

1 Introduction **3**

1.1 Scope 4

1.2 Goals and Motivations 4

1.3 Principal Contributions of this study 5

1.4 Thesis Outline 6

2 Detection and Recognition of Faces **13**

2.1 Face Detection 14

2.1.1 Knowledge-based methods 14

2.1.2 Feature invariant methods 15

2.1.3 Template matching methods 16

2.1.4 Appearance-based methods 17

2.2 Face Recognition 21

2.2.1 Feature-based methods 23

2.2.2 Appearance-based methods 23

2.3 Age Estimation from facial images 26

2.3.1 Anthropometric models 26

2.3.2 Active Appearance Models (AAMs) 27

2.3.3 Age-Manifold models 28

2.3.4 Aging simulation models 28

2.3.5 Feature-Based models 29

2.4 Summary and Proposed Solution 29

3 Gaussian Derivatives as Image Descriptors **35**
3.1 Gaussian Scale Space 36
3.2 The Gaussian Derivatives 36
3.2.1 Steering Gaussian Derivatives 37
3.2.2 Frequency-domain interpretation 39
3.2.3 Filters based in Gaussian derivatives 39
3.2.4 The Gaussian Jet 40
3.3 The Half-Octave Gaussian Pyramid 41
3.4 Summary 43

4 Face Detection with Gaussian Derivatives **47**
4.1 Motivation 47
4.2 Theoretical Background 48
4.2.1 The cascade Architecture 48
4.2.2 Training a cascade of classifiers 49
4.2.3 Detecting faces with a cascade of classifiers 51
4.2.4 Computational Cost of cascade classifiers 51
4.2.5 Evaluation datasets 53
4.2.6 Cascade-Training Setup 56
4.3 Gaussian Derivatives as a feature set 56
4.3.1 Experimental Protocols 57
4.4 Speed-optimized Cascades of Gaussian Derivatives 58
4.5 Experimental Results with non-optimized cascades 61
4.5.1 Sensitivity Results 61
4.5.2 Results on test data sets 61
4.5.3 Results in computational load 63
4.6 Experimental Results with speed-optimized cascades 65
4.6.1 Sensitivity Results 65
4.6.2 Results on test data sets 66
4.6.3 Results on computational load 66
4.7 Discussion and Conclusion 67

5 Histogram-Tensorial Gaussian Representations **73**
5.1 Motivation 73
5.2 Theoretical Background 74
5.2.1 What is a Tensor? 74
5.2.2 Multilinear Principal Component Analysis (MPCA) 75

5.3 Histograms of Binary Gaussian Feature Maps 75
5.4 Tensorial Representations 77
5.5 Fusing Tensors with MPCA 79
5.5.1 Individually Tensors 79
5.5.2 Merged Tensors 80
5.6 Summary 80

6 Face Recognition using Tensors of HBGM 85
6.1 Motivation 85
6.2 Theoretical Background 86
6.2.1 Kernel Discriminative Common Vectors (KDCV) 86
6.2.2 Experimental datasets 86
6.3 Face Recognition Architectures 89
6.3.1 Preprocessing of face images for recognition 90
6.4 Results using First and Second Order Derivatives 90
6.4.1 Results in the FERET face dataset 91
6.4.2 Results in the YALE dataset 92
6.4.3 Results in the Yale B + Extended Yale B Face Dataset 92
6.5 Results using Mag, LoG and γ 92
6.5.1 Computational Cost 93
6.5.2 Performance of y_1, y_2 and y_3 93
6.5.3 Discriminant Capacity of y_T and y_F 94
6.5.4 Results in the FERET face dataset 95
6.5.5 Results in the Yale B + Extended Yale B Face Dataset 96
6.6 Summary and Conclusion 98

7 Age Estimation using HBGM 103
7.1 Motivation 103
7.2 Theoretical Background 104
7.2.1 Relevance Vector Machines (RVM) as regressor 104
7.3 Age estimation using RVM 105
7.3.1 Experimental Datasets 106
7.4 Comparing y_T and y_F performances 107
7.5 Results in the FG-NET database 108
7.6 Results in the MORPH(test) Database 110
7.7 Summary and Conclusion 111

8 Conclusion and perspective **115**
8.1 Principal results 115
8.2 Perspectives 116

A Gaussian Derivatives with the Half-Octave Gaussian Pyramid **119**
A.1 The pyramid algorithm 120
A.2 Gaussian Derivative Feature Calculation 124
A.3 Oriented Derivatives 126
A.4 Scale Interpolated Derivative 127

Bibliography **129**

Introduction

La démonstration d'une méthode rapide et fiable pour la détection de visages par Paul Viola et Michael Jones (Viola and Jones, 2001) a provoqué une révolution dans la vision par ordinateur. Le succès du détecteur de visages de Viola et Jones est le résultat de la combinaison d'un très large ensemble de caractéristiques de l'image avec une cascade de classificateurs linéaires obtenues en utilisant l'algorithme du AdaBoost. Alors que la combinaison du coût algorithmique faible et disposent d'un grand ensemble d'images intégrale est très attractif, la nature binaire des ondelettes de Haar qui en résulte est intuitivement troublante. Traits binaires sont notoirement sensibles aux petites variations dans la position de l'image ainsi que le flou de l'image.

Le point de départ de cette enquête a été d'examiner l'utilisation de multi-échelle Gaussien fonctionnalités dérivées comme une alternative aux ondelettes de Haar comme une caractéristique d'image réglée pour la détection de visage. Tout comme les ondelettes de Haar peut être calculée avec la vitesse O (N) l'algorithme image intégrale, les dérivés de Gauss peut être l'informatique peut être calculée à partir d'un rapide O (N) invariant d'échelle algorithmes pyramide binomiale. Comme avec les ondelettes de Haar, un ensemble potentiellement important de dérivés d'image peuvent être obtenus à chaque position de l'image par des dérivés de l'informatique sur une gamme d'échelles et des orientations et des ordres sur instruments dérivés. Contrairement aux ondelettes de Haar, caractéristiques gaussiennes dérivés ont une sensibilité beaucoup plus faible aux petits changements dans la position et l'échelle des motifs d'image, ce qui permettra éventuellement d'offrir une représentation plus stable. Par ailleurs, la similitude des produits dérivés gaussienne multi-échelle des champs réceptifs dans le cortex visuel suggèrent pertinence comme un ensemble de fonctionnalités objectif image générale, non seulement pour la détection, mais pour le suivi de la reconnaissance et la fonctionnalité.

Nos premières expériences ont démontré que les dérivés multi-gaussienne peut fournir une robustesse améliorée pour faire face à des variations d'orientation, position et l'échelle, tout en offrant des taux de détection des caractéristiques similaires à Haar à un coût réduit de calcul global. Ce succès nous a amené à explorer l'utilisation de fonctionnalités telles pour la reconnaissance faciale et estimation de l'âge. Notre conclusion de ces expériences sont que des caractéristiques multi-échelle Gaussien effectivement fournir un ensemble caractéristique générale et robuste qui sont pratiques pour une variété de problèmes difficiles de vision par ordinateur.

1 Introduction

Chapter Contents

1.1 Scope 4
1.2 Goals and Motivations 4
1.3 Principal Contributions of this study 5
1.4 Thesis Outline 6

THE demonstration of a fast and reliable method for face detection by Paul Viola and Michael Jones (Viola and Jones, 2001) has provoked a revolution in computer vision. The success of the Viola-Jones face detector is the result of the combination of a very large set of inexpensive image features with a cascade of linear classifiers obtained by using the the AdaBoost algorithm. While the combination of low algorithmic cost and large feature set of integral images is highly attractive, the binary nature of the resulting Haar wavelets is intuitively troubling. Binary features are notoriously sensitive to small shifts in image position as well as image blurring.

The starting point for this investigation has been to examine the use of multi-scale Gaussian derivative features as an alternative to Haar wavelets as a image feature set for face detection. Just as Haar wavelets can be computed with the fast O(N) integral image algorithm, Gaussian derivatives can be computed with the fast O(N) scale invariant binomial pyramid algorithm. As with Haar wavelets, a potentially large set of image derivatives can be obtained at each image position by computing derivatives over a range of scales and orientations and derivative orders. In contrast to Haar wavelets, Gaussian derivative features have much lower sensitivity to small changes in position and scale of image patterns, thus potentially providing a more stable representation. Furthermore, the similarity of multi-scale Gaussian derivatives to the receptive fields in the visual cortex suggest suitability as a general purpose image feature set, not only for detection but for recognition and feature tracking.

Our initial experiments have demonstrated that multiscale Gaussian derivatives

can provide improved robustness to variations in face orientation, position and scale, while providing detection rates similar to Haar features at a reduced overall computational cost. This success has led us to explore the use of such features for face recognition and age estimation. Our conclusion from these experiments are that multi-scale Gaussian features do indeed provide a general and robust feature set that are practical for a variety of challenging computer vision problems.

1.1 Scope

Despite the highly developed ability of humans to obtain information from visual observation of faces, facial image analysis remains a very challenging task for computer vision. Over the last few decades, researchers in computer vision have explored a variety of approaches to obtain information from facial images. While most approaches have proven highly sensitive to image noise, illumination and view position, the occasional successes have often signaled the emergence of important new paradigms for computer vision. For example, the demonstration by Kanade (1973, 1977) that relative positions of facial structures such as eyes, mouth and nose could provide discriminant features for recognition was cited as a demonstration of the importance of geometric reconstruction in computer vision. The emergence of the EigenFaces technique of Turk and Pentland (1991b) not only demonstrated the feasibility of using computer vision for recognizing faces from large data bases, but triggered a shift towards appearance based techniques for computer vision. The success of color histogram based methods for face detection led to a widespread investigation of histograms of features (Schiele and Crowley, 1996) ultimately leading to the widely popular SIFT (Lowe, 2004) and HOG (Dalal and Triggs, 2005) feature sets. Of course, the adaptation of Haar wavelets and the AdaBoost learning algorithm for face detection (Viola and Jones, 2001) has driven resurgence of the use of machine learning for computer tasks of all sorts. In summary, facial image analysis has been, and remains, a laboratory for promising approaches for image analysis.

1.2 Goals and Motivations

In this doctoral thesis, we examine the use of multiscale Gaussian Derivatives as an initial image representation for three problems in facial image analysis: face detection, face recognition and age estimation. Our starting intuition has been that such a representation may provide a robust and highly discriminative feature set for facial analysis. The principal result of this investigation is a demonstration that multiscale Gaussian derivatives computed with a Half-Octave Gaussian pyramid can provide a powerful feature set for detection and classification.

Historically, much of the research on facial image analysis has been motivated by applications in security and visual surveillance. The FERET facial recognition benchmark (Phillips *et al.*, 2000) has provided a widely popular competition for highly specialized algorithms, while attracting large amounts of media attention,

publications and research money. Progress in this area has primarily exploited the highly controlled nature of mug-shot face images, although some recent progress has been made for recognition in uncontrolled environments.

Image facial analysis has also been explored in the context of human computer interaction (Crowley and Coutaz, 1996), as an additional perceptual modality for aiding speech (Mcgurk and Macdonald, 1976), as well as a the basis for much research in affective computing (Picard, 1997; Littlewort *et al.*, 2011). However, the unconstrained nature the viewing environments for such applications have tended to hamper progress and inhibit applications.

The success of the Viola Jones method was rapidly translated to widespread use of face detection in digital cameras, and more recently in cell phones and other mobile devices. An enormous variety of applications are currently emerging on mobile devices, including such uses as face tracking for 3D rendering, emotion recognition, augmented reality and visual attention estimation.

Finally, automatic estimation of age from facial images has been explored for applications that include, Age-Specific Human Computer Interaction (ASHCI) (Lanitis. *et al.*, 2004; Ramanathan *et al.*, 2009), security (Ramanathan *et al.*, 2009; Ramanathan and Chellappa, 2006), missing individuals retrieval (Ramanathan *et al.*, 2009; Lanitis *et al.*, 2002), internet access control for minors (Guo *et al.*, 2008b; Lanitis. *et al.*, 2004), surveillance monitoring of alcohol or cigarettes vending machines (Geng *et al.*, 2007; Guo *et al.*, 2008a), appearance prediction across aging (Suo *et al.*, 2010), and targeting of advertising (Guo *et al.*, 2008a).

In most of these domains, the problem of facial image analysis is made difficult by at least five factors: variations in illumination, variations in facial orientation and distance, facial expressions, age variations and occlusions (Abate *et al.*, 2007; Zhao *et al.*, 2003). Many approaches that combine robust feature sets and dimensionality reduction techniques have been proposed to deal with one or a combination of these factors. Thus a technique that can provide robust image description in the presence of these five factors has great potential impact.

1.3 Principal Contributions of this study

This thesis contributes to the scientific research on computer vision by demonstrating new methods for the following three tasks: face detection, face recognition and age estimation from images.

- We propose a new facial image analysis technique that uses the multi-scale Gaussian derivatives as a unique image representation. In general, different images representation are used for different facial analysis tasks (detection and recognition). In this thesis, we demonstrate that the Gaussian scale computed with a Half-Octave Gaussian Pyramid may be used as unique image representation.

- For the task of face detection, we propose the use of a cascade of classifiers using Gaussian derivatives computed with a half-octave binomial pyramid.

In addition, we propose a speed-optimized cascade framework which takes into account the Gaussian derivative's computational complexity and local appearance information as well to select its adequate position in the cascade.

- Use of Gaussian derivatives up to the fourth order are considered in a cascade of classifiers. Despite its high sensitivity to noise, experiments show that inclusion of higher order derivatives improves detection rates.

- We propose a new metric to compute computational load based on the number of requests to the image representation which is more suitable for evaluating feature performance in face detection.

- we perform several experiments for comparing the performance between Gaussian derivative features and Haar-like features when the input image is modified by different transformations such as contrast, noise and rotation. Such transformations are similar to those found in real life applications as video surveillance and biometric.

- we propose a new tensorial representation based in Gaussian derivatives. In this representation LBP is used to build a new image representation and MPCA (Multilinear Principal Component Analysis) is applied for reducing the dimension of the feature space. Two different tensorial frameworks are proposed and its advantages and drawbacks are discussed further in this thesis.

- Finally, we combine different machine learning methods with our tensorial representation for improving results in Face recognition and Age estimation. In this scope, face recognition is addressed as a classification problem using Kernel Discriminative Vectors to improve recognition rates. On the other hand, Age estimation is addressed as a regression problem using Relevance Vector Machines (RVM).

1.4 Thesis Outline

In the following, the content and experimental results of each chapter are summarized.

- **Chapter 2** summarizes references which have been sources of inspiration for different aspects of the thesis. In particular, we discuss three main topics in facial analysis: Face detection, Face recognition and Age estimation.

  In Face Detection, four categories of approaches are presented: Knowledge-based methods, feature invariant approaches, template matching methods, and appearance-based methods. In this scope, we focus on the appearance based methods using cascades of classifiers. On the other hand, we studied different machine-learning methods used in the state-of-the-art for training an efficient cascade of classifiers.

In Face Recognition, we explain appearance-based models, specially those that project the image into a more discriminative space with a suitable dimensionality. This models have been widely used by the face recognition community in the past years.

In Age Estimation from facial images, we review the five different approaches commonly used: Anthropometric, Active Appearance, Age-Manifold, Feature-based and Aging simulation models. In this thesis Feature-based approaches are of special interest.

- **Chapter 3** is devoted to the discussion of different feature sets computed from the Gaussian scale space. In this scope Gaussian derivatives are explained using two different domains, space and frequency. In spatial-domain, the different equations for "steering" the Gaussian derivatives are presented. In frequency-domain, the Fourier transform of the Gaussian derivatives is explained for showing the advantages and drawbacks when higher derivative orders are considered.

  Later in the chapter, we explore the filters based on Gaussian derivatives, specially attention is given to the Gaussian Jet of second order and its norm, Laplacian of Gaussian (LoG) and the Gradient magnitude.

  At the end of the chapter, a highlight of the Half-octave Gaussian pyramid is presented as a fast way to compute Gaussian derivatives. This chapter is complemented by the extensive explanation of the pyramid algorithm described in the appendix A.

- **Chapter 4** applies the Gaussian derivatives for detecting faces using cascades of classifiers. Formally, we present a speed-cascade optimization based in the computational cost of Gaussian derivatives and the captured information necessary to perform an adequate detection process. In this process lower and non computational expensive derivative orders are considered in the first nodes in the cascade where the most highly computational load is presented, the position of derivatives in the cascade is chosen taking into account the detection rate in the current node.

  All the concepts presented in this chapter are supported by experiments performed in the well know MIT-CMU face database and the challenging FDDB face dataset. Experiments have shown the advantages and drawbacks of Gaussian derivatives in face detection. Besides results in computational load and detection rate when different image transformations as rotation, Gaussian noise, contrast and blurring are also conducted to show the advantages of Gaussian derivatives in the face detection process.

  - In the MIT+CMU face data-set results using Gaussian derivatives are competitive or inferior to the state art approaches in face detection, nevertheless this data set is composed of low quality and scanned images that affects the performance of Gaussian derivatives features to detect faces. Results in the FDDB(Face Detection Data Set and

Benchmark) show that Gaussian derivatives features outperform Haar-features in almost 4% of difference in the detection rate. We want to emphasize that FDDB is a challenging data-set with real world conditions (see sections 4.5.2 and 4.6.2).

- In the sensitive test data set, Gaussian derivatives features show a high invariance to rotation and blurring variations, but low invariance to noise presence in the images, this is due to the presence of higher derivative orders (see sections 4.5.1 and 4.6.1).
- In computational load, optimized cascades of Gaussian derivatives features computed using a half-octave Gaussian pyramid exhibit a reduction in computational load (almost 30%) over the non optimized version using the same feature set and the Haar-features computed using the integral images (see sections 4.5.3 and 4.6.3).

- **Chapter 5** develops a tensorial framework based in Histograms of Gaussian Binary Maps for facial analysis. The chapter defines and discusses the different steps to build a tensorial representation taking into account the different possible dimensions as orientation, position and scale. In this chapter, we also consider two different tensorial architectures, the first considers each one of derivative orders as a separate tensor and the second considers the correlation between derivatives when the order is added as supplementary dimension in the final tensor. In addition Multilinear Principal Component Analysis is presented as an algorithm to reduce the dimensions in a tensor without loss of 3-D structure due to vectorization and also as a statistical method for capturing the most discriminative information from each considered dimension in the tensors.

- **Chapter 6** describes as Tensorial representations of Gaussian derivatives (Tensors of HBGM) are applied to the face recognition problem. In this chapter Multilinear Principal Component Analysis is used for reducing the feature space dimension and Kernel Discriminative Common vectors (KDCV) is used to improve the recognition results. In more detail this chapter is divided in three main parts, the first one makes a briefly presentation of KDCV. In the second part only derivatives of first and second order are used in our tensorial representation. This part of the thesis was related with our first results in tensorial representations and only the final vector y_T was used at this time. In the last part of this chapter three well know types of features based in Gaussian derivatives are considered in our tensorial representation: Mag(gradient magnitude), LoG (Laplacian of Gaussians) and γ (the third component of the second local order Gaussian jet norm).

  In all the chapter, three Public available face data-sets (Feret, Yale and Yale B + Extended Yale) are used to validate the approach.

  - Experiments in the FERET face data-set using Gaussian derivatives features of first and second order show that our approach is competitive (similar performances) with others state of the art approaches that use

more complicated feature sets (see section 6.4.1). In addition results in the Yale dataset using Gaussian derivatives features up to the second order in a tensorial fashion outperform other approaches in almost 1% (see section 6.4.2). Finally results in the Yale B + Extended Yale can be observed in section 6.4.3, this results reveals a high invariance of our method to illumination variations.

  - Results in the FERET dataset using Mag(gradient magnitude), LoG (Laplacian of Gaussians) and γ (the third component of the second local order Gaussian jet norm) as well as its computational cost are presented in sections 6.5.4 and 6.5.1 respectively, once again Gaussian derivatives achieve comparative or superior results with other approaches in the state-of-the-art. Besides we compare also the performance of each resulting vector using the two tensorial configurations proposed in this thesis (see sections 6.5.2 and 6.5.3). Finally results in the Yale B + Extended Yale dataset are presented in section 6.5.5.

- **Chapter 7** extends the application of Tensorial representations to the age estimation problem using gaussian derivatives features. In particular, this chapter addresses the problem of age estimation as a regression problem using the vectors y_T and y_F (see section 7.4) as inputs for training a regressor using Relevance Vector Machines. To address the problem of age estimation from faces, we use gaussian derivatives up to the fourth order for getting important aging facial characteristics that can not be described using only Gaussian derivatives of lower order. Two public available datasets (FG-net and MORPH aging datasets) are used to show the quality of the approach to solve this problem. The results are competitive whit the last state-of-the-art methods proposed in the facial analysis field (see sections 7.5 and 7.6).

- **Chapter 8** concludes the principal results and lists perspectives of the thesis.

Détection et reconnaissance de visages

Ce chapitre résume les références qui ont été source d'inspiration pour les différents aspects de la thèse. En particulier, nous discutons de trois thèmes principaux dans l'analyse du visage : Détection des visages, reconnaissance des visages et l'estimation de la vieillesse.

En détection de visages, quatre catégories d'approches sont présentées : les méthodes fondées sur le savoir, la fonction invariante des approches, des méthodes modèle correspondant, et l'apparence des méthodes basées sur. Dans ce cadre, nous signalons spécialement dans les méthodes utilisant les cascades apparence en fonction des classificateurs. En revanche, nous avons exposé différentes méthodes d'apprentissage machine utilisée dans le state-of-the-art pour la formation d'une cascade de classifieurs efficaces.

En reconnaissance des visages, nous expliquons l'apparence des modèles, spécialement ceux. Que le projet de l'image dans un espace plus discriminante avec une dimension appropriée Cette modèles ont été largement utilisés par la communauté de reconnaissance faciale dans les années passées.

En estimation de l'âge à partir des images faciales, nous exposons les cinq différentes approches couramment utilisées : anthropométriques, Apparence Active, l'âge du collecteur, orienté fonction et de modèles de simulation du vieillissement. Dans cette thèse Feature approches fondées sur des intérêts particuliers.

2

Detection and Recognition of Faces

Chapter Contents

2.1 Face Detection **14**
- 2.1.1 Knowledge-based methods 14
- 2.1.2 Feature invariant methods 15
- 2.1.3 Template matching methods 16
- 2.1.4 Appearance-based methods 17

2.2 Face Recognition **21**
- 2.2.1 Feature-based methods 23
- 2.2.2 Appearance-based methods 23

2.3 Age Estimation from facial images **26**
- 2.3.1 Anthropometric models 26
- 2.3.2 Active Appearance Models (AAMs) 27
- 2.3.3 Age-Manifold models 28
- 2.3.4 Aging simulation models 28
- 2.3.5 Feature-Based models 29

2.4 Summary and Proposed Solution **29**

FROM a biological approach, facial analysis is an important processes developed by the human visual system in specific zone of the brain. From birth moment, humans have an innate reflex to attend to faces as a source of information for survival. Facial proportions and expressions are important to identify origin, emotional tendencies, state of health and vital social information.

A problem that seems simple for humans is in fact very challenging for computers. Research to identify what visual information the human visual system uses to represent a face has been conducted in (Valentin *et al.*, 1997; O'Toole *et al.*, 2000; Sinha *et al.*, 2006). Researchers in computer vision have proposed a variety of approaches to extract the same information from facial images acquired under unconstrained conditions such as sensor noise, variations of viewing distance and illumination conditions.

This chapter which reviews the most important advances in facial analysis techniques is organized as follows. In section 4.2, we discuss the face detection process as a first module of an automatic face verification system (see figure 2.1) with relevance in human computer interaction. In section 2.2 an overview of the automatic face recognition process is developed. Facial age estimation is explained in section 2.3. In complement a complete state-of-the-art with the most important approaches in face perception.

2.1 Face Detection

One of the most important techniques that enables human-computer interaction (HCI) is face detection. In general facial analysis algorithms such as face recognition, face alignment, facial expression tracking/recognition, age estimation, head pose tracking and many more, face detection is the first and key step as shown in figure 2.1. It is expected that when computers can really understand the human face, the computers are going to understand people's intentions and the human-computer interaction will be naturally as in human-human interactions.

Many research teams have worked in face detection, and as consequence hundreds of approaches have been published. Some of them have been materialized with success in commercial products as laptops, mobile phones, digital cameras and surveillance systems. Most works in face detection have been well explained in surveys (Yang *et al.*, 2002; Hjelmas and Low, 2001) and recently in (Zhang and Zhang, 2010). Based on these, face detection research could be classified in four categories: Knowledge-based methods, feature invariant approaches, template matching methods, and appearance-based methods. Each one of this categories is explained in the next subsections.

2.1.1 Knowledge-based methods

A first way to detect faces on image is using a set of simple rules that describe with accuracy the human-face proportions from facial images. The rules are provided by a human expert, for example, the center of a facial image has uniform intensity values as well as a considerable intensity difference with the borders and also some morphological considerations like a human-face has always two eyes, a noise and a mouth. A well-know work of this type was proposed by Yang and Huang (1994).

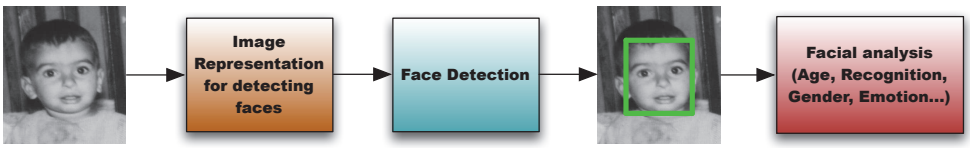


Figure 2.1: *Automatic face verification system*

In their approach, they applied a set of simple rules at different locations in a hierarchy of images with different resolutions as shown in figure 2.2.

The principal inconvenience of this approach is the difficulty to find an effective set of rules. Methods based on this approach also tend to exhibit poor generalization in the detection process due to high variations in pose, illumination and face expression that require a separated set of rules for each case.

2.1.2 Feature invariant methods

The main objective of this methods is to find a set of facial structure features invariant to pose and lighting variation. Examples of such features are the skin color and texture (Ruan and Yin, 2009).

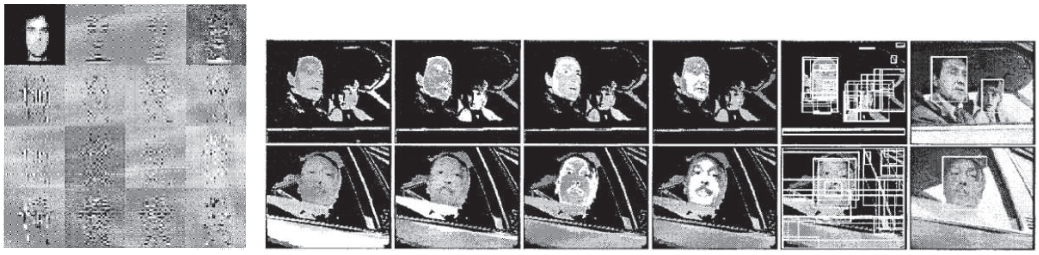
A first method that uses the facial morphological structure is proposed by Han *et al.* (1997). They extracted eye-analogue segments, and choose a set of candidate regions by taking into account facial geometric constraints between eyes, nose, mouth and eye-brows. Finally a neural network is applied over each region to verify if it is a face or not. Their research showed that the most discriminative features in the human face are the eyes; this assumption was corroborated later by Gourier *et al.* (2004).

After facial structure as feature, Amit *et al.* (1998) proposed a two-level classification model, the first level used a spatial configuration of facial edges extracted with a simple edge detector, then a simple detector is trained using Classification And Regression Trees (CARTs). Such detectors retain the most discriminative facial edges and detect frontal-faces.

Color skin is also an important feature in face detection, Garcia and Tziritas (1999) used the skin color as invariant features as did Schiele and Waibel (1995) and Crowley and Berard (1997). First a wavelet decomposition is performed over specific regions represented by the color spaces YCbCr and HSV (figure 2.3a). Restrictions in shape and surface are imposed to discriminate some candidate regions, followed by a probabilistic measure derived from the Bhattacharrya distance is applied to chose the candidate regions that contains a face or not (figure 2.3b).

Sahbi and Boujemaa (2002) computed a set of facial candidate regions using a simple color segmentation process as show in figure 2.4. They detect the most

Figure 2.2: *Hierarchy of resolution images proposed by (Yang and Huang, 1994). Each square cell consists of $n \times n$ pixels in which intensity of each pixel is replaced by the average intensity of the pixels in that cell. (Courtesy of Yang* et al. *(2002))*



(a) (b)

Figure 2.3: *(a) Wavelet decomposition performed over a facial candidate region. (b) Final regions after take into account restrictions and probabilistic metrics (Courtesy of Garcia and Tziritas (1999))*

important candidate region using a neural network trained to recognize human skin color, finally a Gaussian model then is applied to chose the regions that correspond to a face.

While feature invariant methods show an invariance to pose, methods based on skin color are affected by changes in illumination color and intensity, image noise and occlusions. Methods based in facial edges can be affected by shadows, giving some non desired facial edges.

2.1.3 Template matching methods

In Template Matching methods a standard face-template is correlated with candidate regions, and correlation scores are computed individually for each facial component (nose, mouth, eyes etc). A candidate region is validated as a face by aggregating their scores.

Template methods can be illustrated by the method proposed by Heiselet *et al.* (2001). In their method they use a hierarchy of support vector machines trained specifically to detect each facial component. SVMs were trained using a set of synthetic facial images. Finally each facial component is correlated with a template using a simple classifier that verifies spatial organization of each facial feature in the facial candidate. The overall system is presented In figure 2.5

Keren *et al.* (2000) proposed Antifaces, a set of multi-template detectors based in a very simple set of filters. The main idea of this method is that a candidate image region is passed through a sequential set of detectors based on templates. In each detector, the candidate image is correlated by an inner-product with an optimized template to compute the correlation score, used to determine if the candidate region should pass to the next detector or not.

A recent template method was proposed by Hall and Crowley (2004). They used a template computed with a histogram in log-polar space. As first steep a bank of Gaussian derivative filters is applied at multiple scales and locations. A K-means algorithm is then used to detect the most important facial features in multiple clastons.[1] Those are expressed in a gray level image to finally perform the

[1]The word clastons is referred by the authors as the binary images from the clustering process.

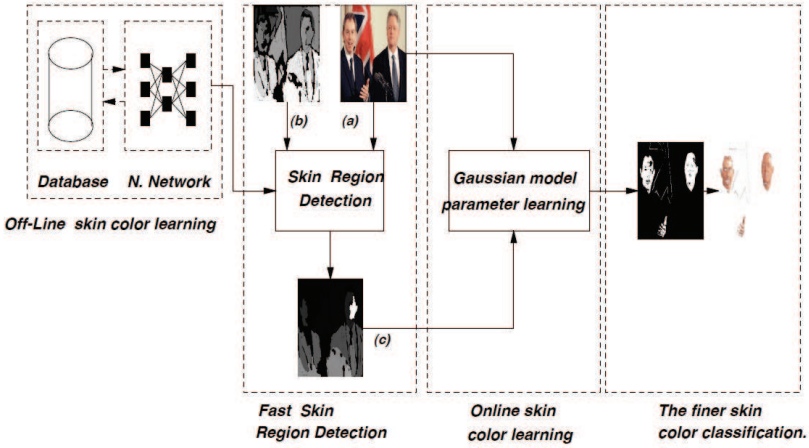


Figure 2.4: *Diagram of skin region selection proposed by Sahbi and Boujemaa (2002) (Courtesy of Sahbi and Boujemaa (2002))*

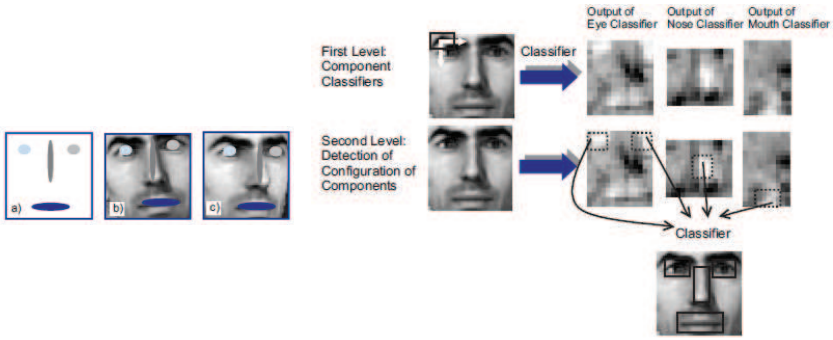


Figure 2.5: *Example of facial template and Overall face detection system proposed by Heiselet et al. (2001) (Courtesy of Heiselet et al. (2001))*

detection process computing the distance between the log-polar histogram in the candidate image and a template log-polar facial histogram. The complete model is summarized in figure 2.6

In spite of the simplicity of this type of method for detection, a good definition of template remains challenging in the case of multi-view face detection because a template is necessary for each face pose.

2.1.4 Appearance-based methods

As was observed in the preceding sections, face detection methods, the definition of a face model is a challenging task that sometimes requires the support of a human expert to define a set of rules or a template that correctly models human facial variations. In this scope, appearance-based methods try to capture the

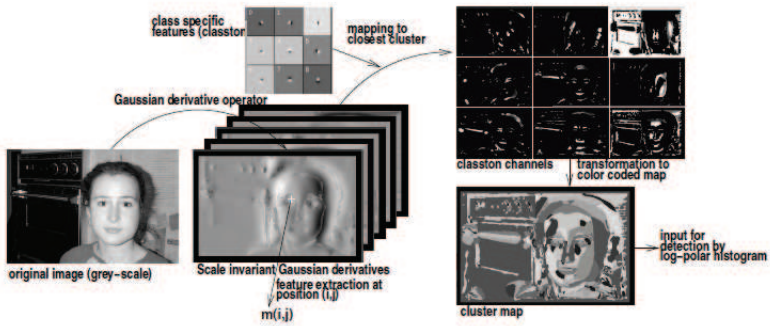


Figure 2.6: *Overall face detection system proposed by Hall and Crowley (2004). (Courtesy of Hall and Crowley (2004))*

most representative variations from a set of example images. To capture such variations in an automatic way without human-intervention, researchers in this field have used machine learning and statistical analysis methods such as Principal Component Analysis, Neural Networks, Support Vector Machines and the well known Adaboost with its variations.

Linear Subspace Methods

The use of Linear Sub-space methods for face detection by Turk and Pentland (1991b) used Principal Component Analysis to compute a set of facial representative images called Eigenfaces from a set of images containing faces and non-faces. Each computed eigenface is clustered taking into account its type of class (face or non-face), then detection process is performed computing the distance between the candidate image region represented as eigenface and each one of computed clusters. Finally, the set of distances computed from a candidate image is defined as "face map" and a face is located in its local minimum.

Distributions functions have been used to construct face detectors. One of the most representative examples from this type was proposed by Sung and Poggio (1998). They used a distribution function to model the differences between faces and non-faces. To do this, each image in the training set is normalized to a dimension of 19×19 pixels and then represented as a vector. Next, each one of those vectors is grouped in six clusters for faces and six for non-faces. In the detection process, two distances are computed between the clusters and the different regions in the candidate image, to finally run the classification process between faces and non-faces with a multilayer perceptron.

Neural Networks

Neural networks have been explored as method to detect faces. A seminal work on this approaches was reported by Agui *et al.* (1992). They developed a hierarchical

model with two levels. In the first level two parallel neural networks take as inputs the intensity values of the candidate region and its filtered values computed with a Sobel filter. In the second level, the values computed by the first level are considered as well as the standard deviation and the ratio between black and white pixels in the binary version of the input image. Detection process is performed based in the output values from the second level.

A first multi-view face detector was proposed by Rowley *et al.* (1998), its method is composed by two consecutive neural networks. The first network is trained to determine the orientation of the input image. The second neural network is trained with frontal-face images of size 20x20 and determines if the input image corresponds to a face or not, taking into account the output information from the first layer.

Garcia and Delakis (2004) have developed a method using a neural-network architecture called a convolutional neural network. Two different types of layers are present in this architecture: convolutional and classification layers. In the convolution layers, a succession of convolutions with a set of Gaussian receptive fields is applied over the input image, once the candidate image has passed across all these layers, the final classification in face or non-face is performed in a neural-network composed by two layers.

Finally in recent work, Osadchy *et al.* (2007) have proposed a method to detect faces and estimate pose based in a convolutional network trained using an Energy-Based model. The convolution neural network is trained for mapping the input images in a low-dimensional manifold, where the different poses are easily separable and parameterized by some typical facial pose parameters (e.g. pitch, yaw and roll). Then the detection process is performed by thresholding the distance between the projected input image and the parameterized manifold.

Support Vector Machines

Another machine learning method widely used in face detection as well as object recognition is Support Vector Machines (SVM). Use of SVM for detecting faces was proposed by Osuna *et al.* (1997). They trained a SVM using images of size 19x19 pixels. In the training set, two different classes are present, faces and non-faces. Each training image is transformed in a vector of 391 components corresponding to original intensity values. The resulting vector provides a feature space for use by a SVM with only 1000 support vectors to classify between faces and non-faces.

Maydt and Lienhart (2002) trained a SVM using as high-dimensional feature space provided by an image decomposition in Haar wavelets (Papageorgiou *et al.*, 1998). The training process is similar to (Osuna *et al.*, 1997). Waring and Liu (2005) used SVM with a feature set composed by spectral histograms from filtered versions of the input image using Gabor filters, gradients and Laplacian of Gaussian(LoG).

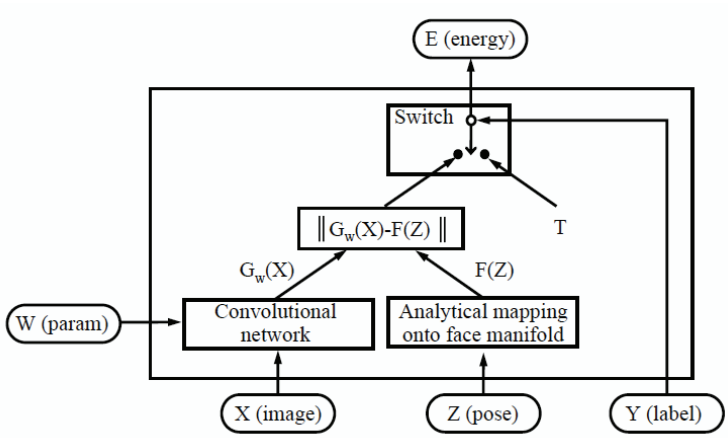


Figure 2.7: *Energy model and neural network architecture proposed by Osadchy et al Osadchy* et al. *(2007) (Courtesy of Osadchy* et al. *(2007))*

Adaboost-based Methods

Adaboost is an iterative machine learning method proposed by Freund and Schapire (1997). Adaboost selects a set of weak classifiers, whose aims to minimize the classification training error on a particular distribution of training samples. At each iteration, adaboost updates the weight of each training sample such that the misclassified samples get more weight in the next iteration.

A real time and highly effective face detector was proposed by Viola and Jones (2001, 2004). They have demonstrated that a rapid and robust object detector can be constructed using a cascade of linear classifiers learned with Adaboost. As a feature set they used Haar-like features (Papageorgiou *et al.*, 1998), such features can be computed quickly and efficiently with an Integral image by a simple sum of values in a rectangle subset of a grid. A very large number of descriptors (180 000 for a 24×24 image) can be computed with such image representation.

In face detection, the cascade of linear classifiers is successively applied to all image sub-windows. The first layer has a small number of weak classifiers that reject a pre-defined percentage of negative sub-windows and detect nearly 100% of the positive sub-windows in the image. The next layer is then trained to reject the same percentage of negative sub-windows and detect nearly 100% of positive sub-windows using the sub-windows that were improperly classified by the previous layer. This procedure is repeated to provide a cascade of classifiers that increasingly concentrate on a reduced number of difficult sub-windows. This technique allows an improvement in the detection speed with excellent detection and false positives rates. More details about face detection with cascades of linear classifiers are explained in chapter 4.

Inspired by the method of Viola and Jones, many researchers have explored new strategies to improve cascade's performance. Three approaches have been used to improve detection performances:

- Training cascades with a new set of features to improve linear classifiers and deal with illumination problems. Some extensions derive from the original haar features proposed by Viola and Jones are rotated-haar features (Lienhart *et al.*, 2003) (figure 2.8a) and Haar-like features for multi-view face detection (Li *et al.*, 2002; Xiao *et al.*, 2004) (figure 2.8b). Feature sets based in Local Binary Pattern (LBP) and Gabor features are also popular, some examples of them are Haar Local Binary Patterns (Roy and Marcel, 2009), Locally assembled Binary features (Yan *et al.*, 2008a), Anisotropic Gaussian Filters (Meynet *et al.*, 2007) (figure 2.9a), Gabor features (Huang *et al.*, 2005b) (figure 2.9b), Gabor Features computed with an Integral Image (Xiaohua *et al.*, 2009) and Region Covariance (Tuzel *et al.*, 2006; Pang *et al.*, 2008).

- Optimizing the cascade learning methods. Chen and Chen (2008) proposed meta-stages trained with SVMs to improve detection average and discard more quickly negative examples. Brubaker *et al.* (2008) improved the cascade's learning using the characteristic points in the Receptive Operation Curves (ROC). Wu *et al.* (2008) proposed Linear Asymmetric Classifiers (LAC) to deal with the asymmetries problems for training cascades of classifiers in face detection.

- Improving adaboost as feature selector for detecting faces. Eigenboosting (Grabner *et al.*, 2007), Kullback-Leibler boosting (Liu and Shum, 2003), Vector boosting (Huang *et al.*, 2005a), Floatboost (Li and Zhang, 2004) and FFS (Forward Feature Selection) Wu *et al.* (2008) are examples of improvements in the original adaboost method.

Face detection by appearance is the most used type of method by the computer vision community, this is due to its superior performances compared to other methods and the high improvement in computational power, which has made possible to develop complicated machine learning algorithms in real time.

We will return to the problem of face detection with cascades of linear classifiers and adaboost in chapter 4.

2.2 Face Recognition

Automatic face recognition has emerged as an active field of research driven by the promise of applications in security surveillance, access control, human-computer interaction, and many other domains. Face recognition includes two different topics:

- **Face Identification:** An unknown image of a face or probe image is compared against every record in a database (Gallery) and the face identification system attempts to answer the question "Who is X?". This type of comparison is called a "one-to-many" search (1:N).

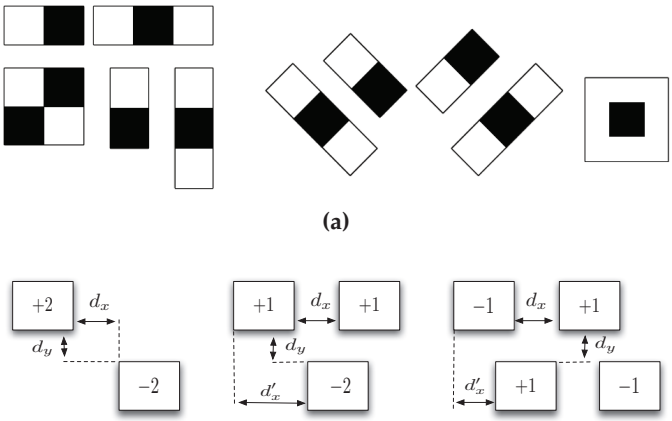


Figure 2.8: *Haar-like feature sets.(a)Original Feature set proposed by Viola and Jones (2001) and oriented Haar-like features proposed by Lienhart* et al. *(2003). (b)Sparse features represented in granular space proposed by Li* et al. *(2002)*

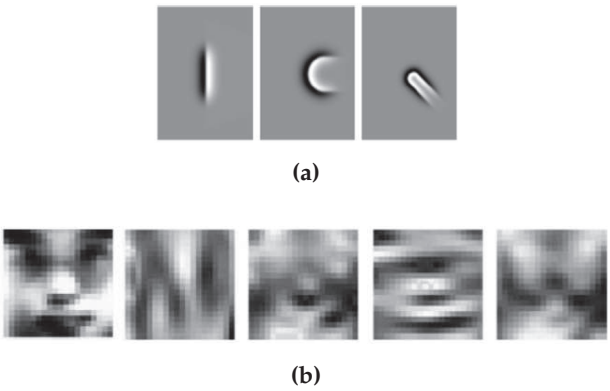


Figure 2.9: *(a)Anisotropic Gaussian Filters proposed by Meynet* et al. *(2007) (Courtesy of Meynet* et al. *(2007)). (b)Facial images based in Gabor filter as proposed by Huang* et al. *(2005b)(Courtesy of Huang* et al. *(2005b))*

- **Face Verification**: A claimed identity must be validated based on the image of a face, and either accepting or rejecting the identity claim, in this case the system try to answer the question "Is this X?". This type of comparison is called a "one-to-one" search (1:1).

Face recognition systems have been generally accepted by the public. However, limited reliability of face recognition systems continues to inhibit its widespread deployment.

Performances of an automatic face recognition system can be affected by five factors: illumination variations, pose changes, facial expression, age variations and occlusions (Abate *et al.*, 2007; Zhao *et al.*, 2003). Many approaches that combine robust feature sets and dimensionality reduction techniques have been proposed to deal with one or a combination of these factors.

As illustrated by (Abate *et al.*, 2007; Zhao *et al.*, 2003) face recognition methods are generally classified in three different types of approaches: Feature-based, Appearance-based and Discriminitive-space-based methods.

2.2.1 Feature-based methods

In this methods, important facial features as eyes, nose, mouth are first extracted taking into account their spatial position and appearance as well, then this information is used in a structured-based classification.

A very early work (perhaps the first) feature-based classification system was developed by Kanade (1977). This system used fiducial-facial angles and distances between the eye corners, mouth extrema, chin-top and nostrils.

In a more recent work proposed by Ashraf *et al.* (2008), facial information is modeled as a set of patches computed from different viewpoints (see figure 2.10). Patches are initially aligned by a data-driven framework and then compared with a aligned version of the gallery set using a probabilistic model.

Feature based methods consider geometric shape of face that could be important in the recognition process. The inconvenience of this approach is that facial features have to be reliable extracted. In many cases, this can be more challenging than face recognition process itself.

2.2.2 Appearance-based methods

This kind of methods considers the whole facial image as input to the facial recognition system. Usually facial information is considered as only intensity values or described using a set of robust features as Gabor filters, color spaces etc. Finally facial information is projected into a more discriminative space, where the face recognition process can be performed.

A representative approach that uses pixel intensity values as facial information was developed by Turk and Pentland (1991a), they proposed a facial image representation computed with Principal Components Analysis. Such representation encodes facial appearance by a set of images called eigenfaces. First a set of training images is vectorized to obtain a matrix of data where each column corresponds to an example in the training set. Finally Principal Component Analysis is performed using this matrix and eigenfaces are computed from the resultant eigenvectors. An example of Eigenfaces is shown in figure 2.11

A two dimensional version of PCA was proposed by Yang *et al.* (2004), the main difference between eigenfaces and this method is that facial images do not need to be transformed in vectors and the covariance matrix necessary for computing PCA is constructed directly using the original images. The covariance matrix is calculated taking into a count the variations on each dimension separately.

He *et al.* (2005), proposed Laplacianfaces. The main difference between Laplacianfaces and Eigenfaces is that Laplacianfaces preserves the local structure by

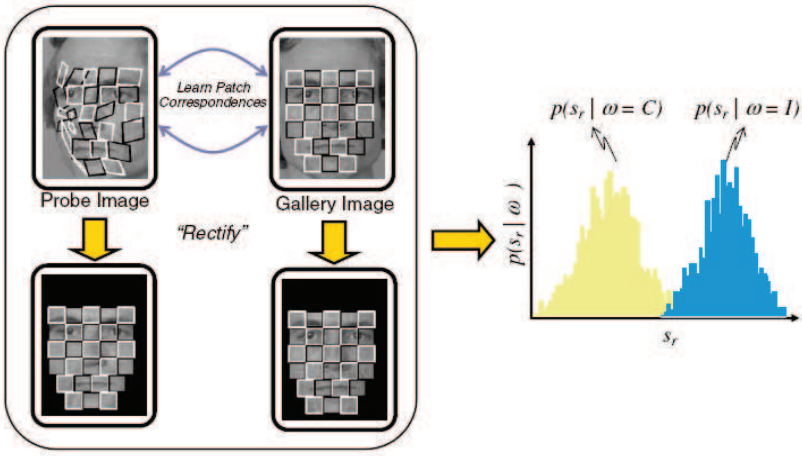


Figure 2.10: *Viewpoint face recognition by patches proposed by Ashraf* et al. *(2008). (Courtesy of Ashraf* et al. *(2008)). The patches from the probe image are compared with the gallery set using a probabilistic model*

an adjacent graph and uses Laplacian Beltrami operator to find the eigenvectors whose preserves 3D structure of the original input images.

The use of original image intensities in face recognition is easy to implement and can be adapted to work in real time however such representations are very sensitive to illumination changes and noise in the image, which are normally present in real-world conditions. To avoid such problems, new facial representations that are invariant to illumination conditions and image noise have been proposed by the object recognition community. Many of the most successful representations in face recognition uses Gabor filters combined with LBP (Local Binary Patterns).

In face recognition field, Local Binary Pattern (LBP) has been proposed by Ojala *et al.* (1996). An example of a LBP is shown in figure 2.12. LBP assigns a label to each pixel of the image by thresholding the 3×3 neighborhood of each pixel with the center pixel value and considering the result as a binary or decimal number. This operator encodes a set of facial micro-patterns from the neighborhood appearance.

Other facial representation that uses Gabor filters is developped by Zhang *et al.* (2005). They propose Histogram Sequence (LGBPHS) that combines the magnitude part of Gabor feature with the LBP operator (Ahonen *et al.*, 2006).

In a more recent publication, Zhang *et al.* (2007) have explored encoding the face image with a Global Gabor Phase Pattern (GGPP) and a Local Gabor Phase Robust Pattern (LGPP). These features are computed taking into account variations in orientation for the Gabor wavelets at a given scale (or spatial frequency).

Tan and Triggs (2007) have explored a fusion of Gabor and LBP features to improve recognition in complicated illumination conditions. First Gabor filters with different scales and orientation are applied to the input images, at the same time LBP is also applied, to finally combine them by a kernel method to improve recognition rates.

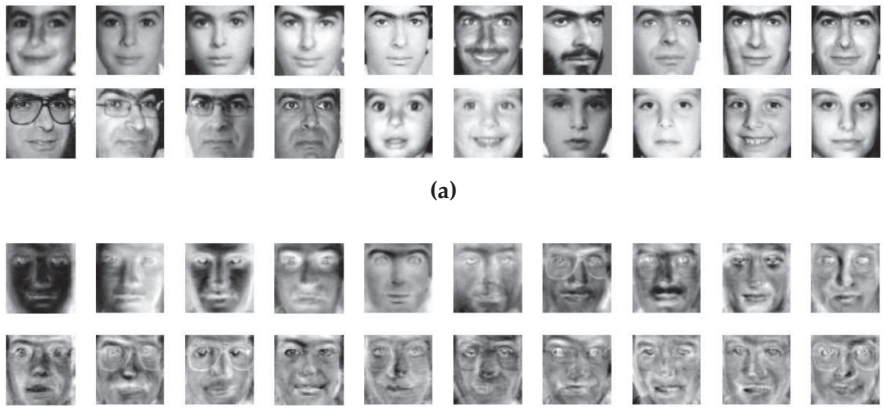


Figure 2.11: *Example of eigenfaces proposed by Turk and Pentland (1991a). (a) Original dataset (b) First twenty eigenfaces from the original dataset.*

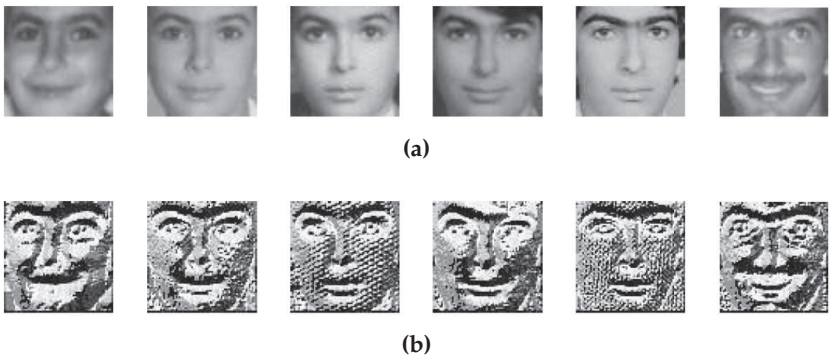


Figure 2.12: *Local Binary Pattern proposed for recognizing faces by Ahonen* et al. *(2006). (a) Original dataset (b) Result images after applying LBP operator in the original dataset.*

Discriminative-space methods

Appeareance-based methods use higher-dimension feature spaces where not all the dimensions are necessary in the face recognition process and made it not suitable for real time applications. A common approach to deal with this problem is to project the image into a more discriminative space with a suitable dimension for fast face recognition. Another approach is to project the original image space to an implicit higher dimensional feature space where the recognition process could be considered as linearly separable. Some examples of discriminative-space methods are Gabor-based kernel PCA (Liu, 2004), Discriminative Common Vectors (Cevikalp *et al.*, 2005, 2006), Kernel Locality Preserving Projections with Side Information (KLPPSI) (An *et al.*, 2008), MLASSO (Pham and Venkatesh, 2008) and Volterrafaces (Kumanr *et al.*, 2009)(see figure 2.16)

We find two main disadvantages in the preceding approaches. The first is that they use computationally expensive features such as Gabor wavelets. The second problem is that dimensional reduction techniques operate over a feature space of

one or two-dimensions. In the case of higher order space feature, this space must be reshaped into vectors. Vectorization breaks the natural structure and correlation in the original feature space (Lu *et al.*, 2008).

2.3 Age Estimation from facial images

Human faces are an important non-verbal source of information for human interaction. In addition to expressions of emotion, the human face communicates gender, ethnic origin and age Ramanathan *et al.* (2009).

Automatic estimation of age from facial images has been used for: Age-Specific Human Computer interaction (ASHCI) (Lanitis. *et al.*, 2004; Ramanathan *et al.*, 2009), security (Ramanathan *et al.*, 2009; Ramanathan and Chellappa, 2006), missing individuals retrieval (Ramanathan *et al.*, 2009; Lanitis *et al.*, 2002), internet access control for minors (Guo *et al.*, 2008b; Lanitis. *et al.*, 2004), and surveillance monitoring of alcohol or cigarettes vending machines (Geng *et al.*, 2007; Guo *et al.*, 2008a), appearance prediction across aging (Suo *et al.*, 2010), and targeting of publicity (Guo *et al.*, 2008a).

Despite the number of potential applications, automatic image-based age estimation remains a challenging problem. Compared with other facial variations, aging effects are very dependent on genetics (Guo *et al.*, 2008b), life style, location of residence (Guo *et al.*, 2009) and weather conditions (Lanitis *et al.*, 2002). Furthermore, males and females age differently, and the apparent effects of aging are often masked by makeup and facial accessories (Suo *et al.*, 2010). Accommodating the influence of individual differences to provide a general method for estimating age based on facial images remains an open problem.

Most automatic image-based age estimation systems are composed by combining two components (Lanitis *et al.*, 2002): an image representation and an age estimation process. The age estimation process can be formulated as a multi-classification problem (Lanitis. *et al.*, 2004) where each age is considered a separate class, a regression problem where an approximative age-function is computed from a set of training images (Guo *et al.*, 2009, 2008a) or an hybrid version that combines classification and regression methods (Guo *et al.*, 2008b).

For image representation, five different approaches are commonly used: Anthropometric, Active Appearance, Age-Manifold, Feature-based and Aging simulation models.

2.3.1 Anthropometric models

These models use cranio-facial information to determine an approximate age. An example of this type of models is proposed by Kwon and Vitoria Lobo (1999). They categorize ages in three different types: babies, young adults and seniors. Facial features as eyes, nose, mouth, chin, virtual-top of the head and the sides of the face are located, then using these positions a set of ratios is computed to distinguish between the three categories above mentioned. The main inconvenient with these

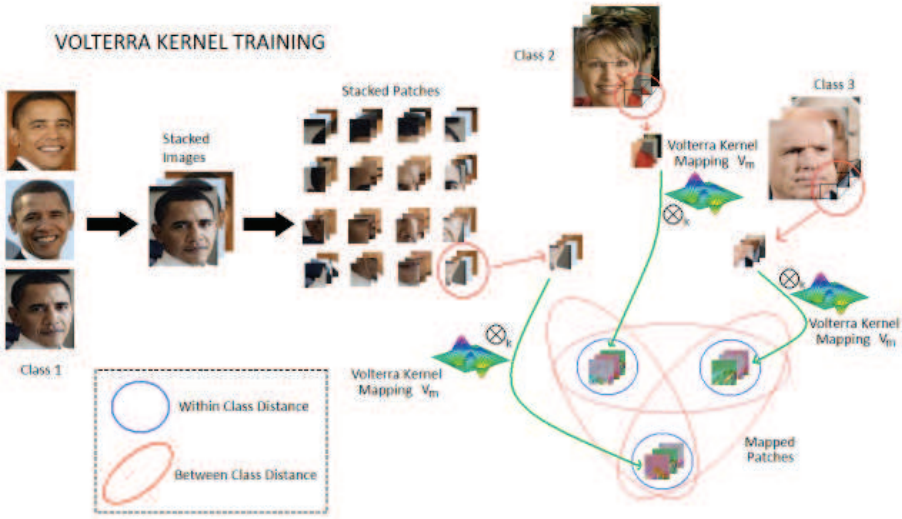


Figure 2.13: *Training and patch images used by Kumanr et al. (2009) to project the original feature space in other more discriminative space using volterra kernels(Courtesy of Kumanr et al. (2009))*

methods is the imprecision of the age estimation process (more categorization than estimation) due to facial feature localization. In addition these methods do not use skin information (texture information) useful for distinguishing between adulthood and old ages.

2.3.2 Active Appearance Models (AAMs)

Active Appearance Models (Cootes *et al.*, 1998) are constructed from a set of training images with hand-made landmarks. Such landmarks are combined with the intensity image values to learn a statistical model that takes into account shape and texture. Originally AAMs have been proposed for matching tracking and recognition. More recently they have been widely used for an image analysis of aging. For example, Geng *et al.* (2007) have described the AGing pattErn Subspace (AGES) that uses AAMs as face models to create sequences of individual aging images sorted in time order to make a representative subspace. For an unseen face image, age is determined by the position of its projection in the subspace that can reconstruct the facial image with minimum reconstruction error as show in figure 2.14.

AAMs models have been widely used in age estimation from facial images, but its principal inconvenience is the use of a hand-made marked image database to compute the model.

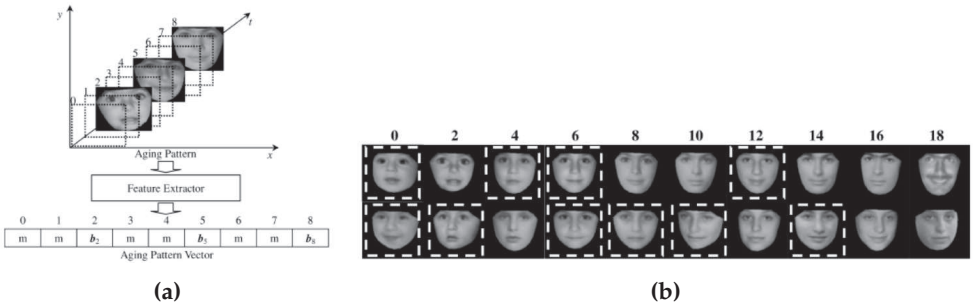


Figure 2.14: *AGing pattErn Subspace (AGES) proposed by Geng et al. (2007). (a) Feature extraction from a time-ordered sequence of facial images. (b) Reconstruction of empty sub-space position to estimate age. (Courtesy of Geng et al. (2007))*

2.3.3 Age-Manifold models

Age-manifold models project facial images in a low-dimensional space in where facial changes due to age can be determined with a higher precision than the original space. For example, Guo *et al.* (2008a) proposed to learn an optimal-subspace using an adequate space projection algorithm. In their experiments, AAMs were used as an image representation and three different projection algorithms (Principal Component Analysis, Locally Linear Embedding and Orthogonal Locality Preserving Projection (OLPP) (Cai *et al.*, 2006)) were compared to compute the optimal age manifold. Age is estimated in the optimal manifold using an adjusted version of support vector machines as regresor.

2.3.4 Aging simulation models

With an Age simulation model, a statistical model is applied to an input image to simulate its complete facial aging process. Scherbaum *et al.* (2007) learned a 3-D aging-face-model from 3D scans of teenagers and adults using support vector regression (figure 2.15). This model considers shape and texture to apply the aging transformation whereas their model is well suited to children and adults but does not work well for older people.

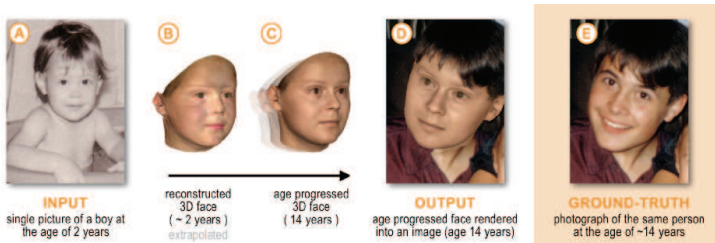


Figure 2.15: *Age simulation example using the model proposed by Scherbaum et al. (2007). A 3-D face model of a boy is transformed into a face at an appropriate target age, then the image is rendered in an appropriate image-background. (Courtesy of Scherbaum et al. (2007))*

Suo *et al.* (2010) have described the use of a Markov process using a sparse graph representation trained with a large annotated face dataset. In a first level of graph, this method describes the appearance of the face and the hair. In a second level facial details are refined using an AAM (Active Appearance Model) trained with 90 landmarks. In the third and last level, six different wrinkle zones are modeled using Gabor wavelets and geometrical position.

2.3.5 Feature-Based models

Feature-based models describe facial aging structure with a set of discriminant texture features. For example Guo *et al.* (2009), developed an age estimation system based in two consecutive bio-inspired units. The first type of units S_1 corresponds to receptive fields at the mammalian visual cortex (cells of Hubbel and Wiesel). These are modeled using a bank of Gabor filters with different orientations and bandwidths. A second set of units called C_1 corresponding to cortical complex cells which are modeled using a non-linear operator applied over the S_1 units. Finally PCA is applied to reduce the feature-space dimension.

Three main advantages of feature-based models are:

- Their invariance to illumination variations and noise in the images.
- Absence of facial-landmarks in the training set used to build the model (Only textural information is used).
- Age-Manifolds models can be used in addition to Feature-based models to improve age estimation process.

The main inconveniences of these methods are

- The required large aging database that can be hard to collect.
- The high computational cost of the filter bank used to compute the facial model.

2.4 Summary and Proposed Solution

Without giving a comprehensive overview, the chapter summarized facial analysis algorithms which are of special interest for this thesis. Face detection, face recognition and age estimation have been emphasized as important modules used in Human-Machine Interfaces (HMI), security systems and law enforcement applications.

In face detection, approaches based in appearance, specially those that uses cascades of simple classifiers have showed a superior performance in terms of speed and performance compared with others in the literature. In this scope, Haar-like features have been widely used by their fast computation with an integral

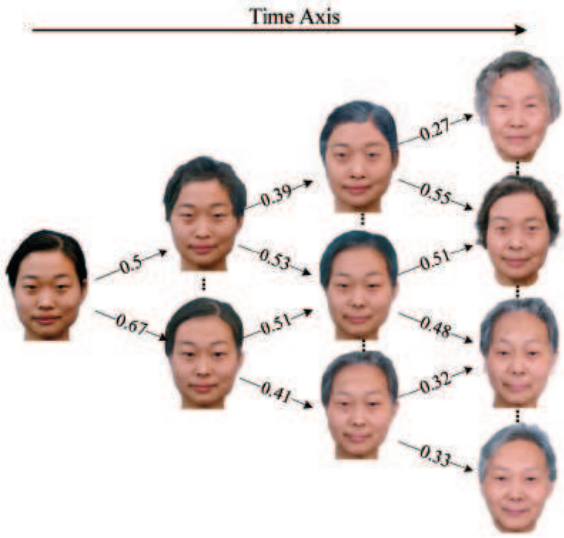


Figure 2.16: *Age simulation example using the model proposed by Suo et al. (2010). An input image is progressively transformed in a set of unseen facial ages. (Courtesy of Suo et al. (2010))*

image representation, besides these are not robust to handling faces under extreme light conditions and their use in real-word conditions requires a normalization by the intensity covariance of each test window.

Many researches have focused their researches in finding new types of feature sets robust to such variations without applying special-corrections of illumination. Following this line, in this thesis we propose to use cascade of simple classifiers using Gaussian derivatives computed with a half-octave pyramid. In addition, we propose Heterogeneous-cascade of classifiers as a new cascade architecture that takes into account the Gaussian derivatives's computational complexity and local appearance information as well to select its adequate position in the cascade.

In face recognition and age estimation, special attention has been fixed in appearance-based models. As it was described above, image representations used in appearance-models must be able to capture discriminative facial information which is important to distinguish a subject's identity and infer his/her age. To address this problem we propose to use a new feature tensorial representation based on binary Gaussian feature maps. This representation retains multi-dimensional spatial structure used to compute the facial representation. In addition, the binary Gaussian feature maps used in the tensors construction can be computed with a half-octave pyramid, which unifies the complete facial analysis using a unique computer vision tool as shown in figure 2.17.

The following chapters explain our solutions. In chapter 3, we present a theoretical introduction of Gaussian derivatives features. In chapter 4, we present the cascade of simple classifiers using Gaussian derivatives features. In addition, chapter 4 describes a new framework to train speed-optimized cascades of classifiers. In chapter 5, a theoretical explanation of the Histogram-Tensorial

Gaussian representations is exposed in details. Finally in chapter 6 and chapter 7, tensorial representations are applied to the face recognition and the age estimation respectively.

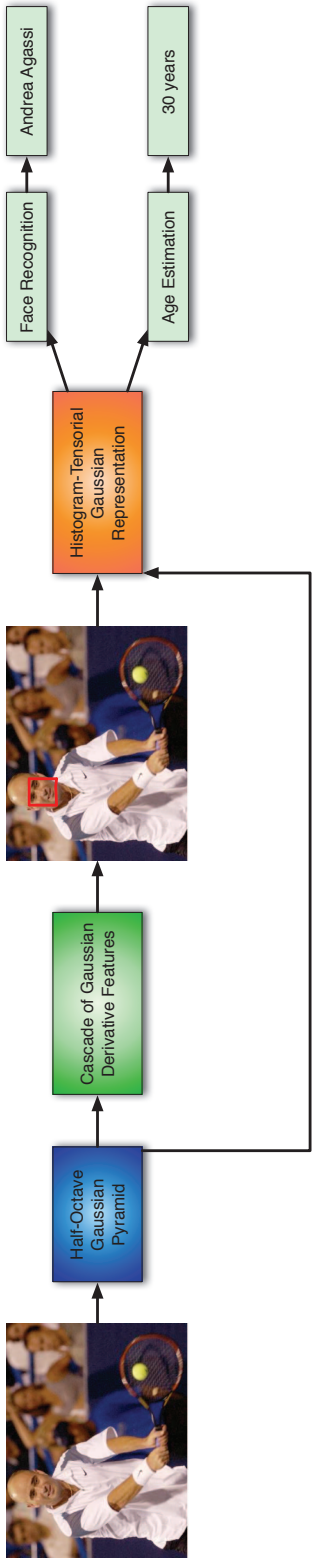


Figure 2.17: *Unified facial analysis architecture proposed in this thesis*

Les Dérivés Gaussiens comme descripteurs d'images

Ce chapitre est consacrée à la discussion des différents caractéristiques calculées à partir de l'espace d'échelle gaussienne. Dans cette aspect, Gaussien dérivés sont expliquées en utilisant deux domaines différents, le domaine de l'espace et le domaine de la fréquence. Dans le domaine de l'espace, les différentes équations pour roter les dérivés gaussiens sont présentées. Dans le domaine fréquentiel, la transformée de Fourier des dérivés gaussiennes est expliquée pour montrer les avantages et les inconvénients lorsque les dérivés d'ordre supérieur sont considérés.

Plus tard dans le chapitre, nous explorons les filtres basés sur les dérivés Gaussianes, spécialement le Jet gaussienne de second ordre et sa norme, le Laplacien de Gaussienne (LOG) et la magnitude du gradient.

A la fin du chapitre, nous présentons la pyramide des demi-octave gaussienne comme un moyen rapide de calculer les dérivées gaussiennes. Ce chapitre est complété par l'explication détaillée de l'algorithme de la pyramide décrite dans l'annexe A

3

Gaussian Derivatives as Image Descriptors

Chapter Contents

3.1 Gaussian Scale Space **36**
3.2 The Gaussian Derivatives **36**
3.2.1 Steering Gaussian Derivatives 37
3.2.2 Frequency-domain interpretation 39
3.2.3 Filters based in Gaussian derivatives 39
3.2.4 The Gaussian Jet 40
3.3 The Half-Octave Gaussian Pyramid **41**
3.4 Summary **43**

THE choice of feature set is an important part of the design of any recognition system. Feature sets should be selected to accommodate factors such as object classes under consideration, the sensor (camera) characteristics and the application scenario (indoor/outdoor). In this scope, human faces belong to an special class of 3-D objects that need a set of features robust to variations due to factors as illumination, pose, facial expression, aging, etc.

In this thesis, we explore a facial analysis framework that uses Gaussian derivatives as a local facial feature which can be calculated locally and robustly with respect to image noise, image blur and scale changes. Feature spaces calculated from Gaussian derivatives are widely used in invariant object recognition (Yokono and Poggio, 2004; Schiele and Crowley, 2000), face recognition (Wright and Hua, 2009), image tracking and scene reconstruction (Lowe, 2004; Tola *et al.*, 2009; Winder and Brown, 2007).

This chapter is organized as follows, section 3.1 introduces the Gaussian Linear Scale space. An overview of Gaussian derivatives up to the fourth order, their "steerability" with respect to the image plane and their analysis in the frequency space will be presented in section 3.2, in the same way Gaussian local jet, second

local Gaussian jet norm and basic image features will be explained as well. Finally theoretical analysis of the half-octave Gaussian pyramid is presented in section 3.3.

3.1 Gaussian Scale Space

The term Scale Space was introduced by Witkin (1984) to describe the blurring properties of one dimensional signals. Koenderink and van Doom (1987) applied this concept to images using the assumptions of causality, isotropy and homogeneity for revealing that the scale space must be essentially governed by the isotropic diffusion equation $\frac{dI}{d\sigma} = c\nabla^2 I$ which shows that many physical phenomena can be described using the Gaussian kernel:

$$G(x, y, \sigma) = \frac{1}{2\sigma\pi} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.1}$$

Where σ is the size of the support in terms of the second moment (or variance).

Crowley and Stern (1984); Crowley and Sanderson (1987); Burt and Adelson (1983) present the first notions for computing the scale space using a pyramidal representation and finally Lindeberg (1994) formalized the concept of the discrete Gaussian scale space.

Following the above mentioned, the Gaussian scale space can be computed as follows:

$$I(x, y, \sigma) = G(x, y, \sigma) * I(x, y) = \frac{1}{2\sigma\pi} e^{-\frac{x^2+y^2}{2\sigma^2}} * I(x, y) \tag{3.2}$$

Where $I(x, y, \sigma)$ is the Gaussian scale-space representation of the image $I(x, y)$ and "$*$" is the convolution operator. An example of scale space is showed in figure 3.1.

3.2 The Gaussian Derivatives

In Neuroscience, the classical receptive field of a sensory neuron is a region of space in which the presence of a stimulus will alter the firing of that neuron. For mammals, receptive fields have been identified for neurons of the auditory system, the somatosensory system, and the visual system. Young *et al.* (2001) have reported that receptive fields in the visual cortex can be well modelled using Gaussian derivative operators up to fourth order.

To describe Gaussian derivatives, we introduce a particular notation which will be used in the next chapters of this thesis. Let $\vec{v}(\theta) = (\cos(\theta)\ \sin(\theta))$, be the directional vector that describes the desired orientation θ for a Gaussian derivative of nth order. In addition, we define the x-axis parallel to $\vec{v}(0^\circ)$, which corresponds to $\theta = 0$. The y-axis is defined by $\theta = 90^\circ$ and is therefore parallel to $\vec{v}(90^\circ)$.

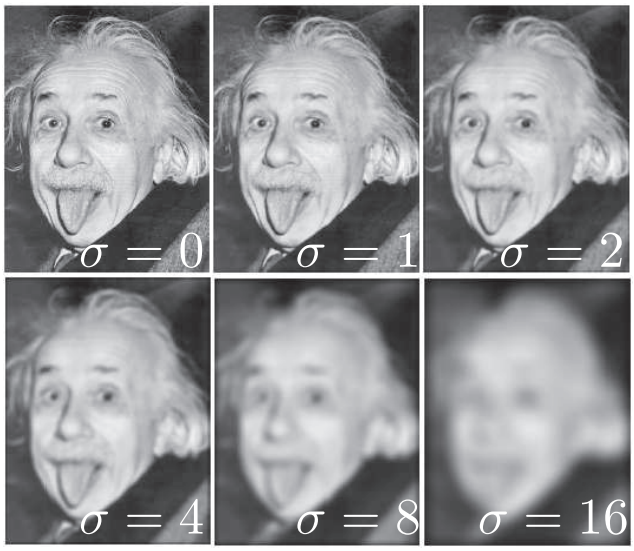


Figure 3.1: *Gaussian scale space representation at different values of σ. Image details are eliminated when the scale value increases. (image from* `http://th.physik.uni-frankfurt.de/~jr/physlist.html`*)*

Following this notation, Gaussian derivatives of nth order at any orientation θ are described by:

$$G_{n,\theta}(x, y, \sigma) = \frac{\partial^n}{\partial^n \vec{v}} G(x, y, \sigma) \tag{3.3}$$

3.2.1 Steering Gaussian Derivatives

Freeman and Adelson (1991) have shown how a basis set of Gaussian derivatives can be "steered" to a desired orientation by weighting the derivative terms with the appropriate sine and cosines terms. This basis set is defined as follows:

$$G_{x^m y^n}(x, y, \sigma) = \frac{\partial^{m+n}}{\partial^m x \partial^n y} G(x, y, \sigma) \quad m, n \in \mathbb{Z} \geq 0 \tag{3.4}$$

where m and n correspond to derivative orders in x and y axis respectively, an example of these basis is shown in figure 3.2.

Consequently, Gaussian derivatives up to fourth order at any orientation θ, expressed in equation 3.3 can be defined in terms of their basis expressed in equation 3.4 using Freeman and Adelson's method as follows:

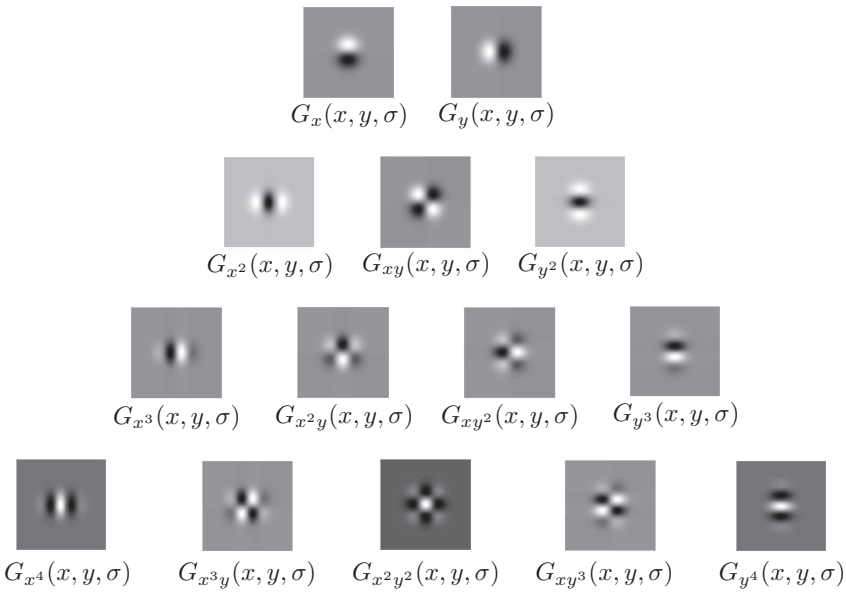


Figure 3.2: *Impulse responses for steering basis of Gaussian derivatives computed at* $\sigma = \sqrt{2}$

$$
\begin{aligned}
G_{1,\theta}(x,y,\sigma) &= \cos(\theta)G_x(x,y,\sigma) + \sin(\theta)G_y(x,y,\sigma) \\
G_{2,\theta}(x,y,\sigma) &= \cos^2(\theta)G_{x^2}(x,y,\sigma) - 2\sin(\theta)\cos(\theta)G_{xy}(x,y,\sigma) \\
&\quad + \sin^2(\theta)G_{y^2}(x,y,\sigma) \\
G_{3,\theta}(x,y,\sigma) &= \cos^3(\theta)G_{x^3}(x,y,\sigma) - 3\sin(\theta)\cos^2(\theta)G_{x^2y}(x,y,\sigma) \\
&\quad +3\sin^2(\theta)\cos(\theta)G_{xy^2}(x,y,\sigma) - \sin^3(\theta)G_{y^3}(x,y,\sigma) \\
G_{4,\theta}(x,y,\sigma) &= \cos^4(\theta)G_{x^4}(x,y,\sigma) - 4\cos^3(\theta)\sin(\theta)G_{x^3y}(x,y,\sigma) \\
&\quad +6\sin^2(\theta)\cos^2(\theta)G_{x^2y^2}(x,y,\sigma) - 4\cos(\theta)\sin^3(\theta)G_{xy^3}(x,y,\sigma) \\
&\quad + \sin^4(\theta)G_{y^4}(x,y,\sigma)
\end{aligned}
\tag{3.5}
$$

Gaussian derivatives of first order capture information about changes of the surface normal and measure the intensity of edges. The second order Gaussian derivatives are good descriptors for image features such as bars, blobs and corners. Higher order Gaussian derivatives are more sensitive to image noise and only provide useful information in cases where the second order derivatives are strong. On the other hand, they are highly selective to specific-local structures in the image and can be useful to detect and recognize faces and to describe small facial changes due to aging.

3.2.2 Frequency-domain interpretation

Gaussian derivatives are not only used by their selectivity to a specific orientation and frequency, to show this in this section, Gaussian derivatives will be analyzed using the frequency domain. For the sake of simplicity and without loss of generalization 1D Gaussians derivatives will be considered.

The Fourier transform of the Gaussian in the x-axis is written as follows:

$$\mathcal{G}\left(\omega_x, \sigma\right) = \mathcal{F}\left\{G(x,\sigma)\right\} = e^{-\frac{\sigma^2\omega_x^2}{2}} \tag{3.6}$$

where ω_x is the frequency variable. From equation 3.6 and taking into account Fourier transform properties, it is possible to define the Fourier transform for the Gaussian derivatives:

$$\begin{aligned} \mathcal{G}_{x^{n_x}}\left(\omega_x, \sigma\right) &= \mathcal{F}\left\{\frac{\partial^{n_x}}{\partial x^{n_x}} G(x,\sigma)\right\} \\ &= \left(-\jmath\omega_x\right)^{n_x} \mathcal{F}\left\{G(x,\sigma)\right\} \end{aligned} \tag{3.7}$$

The Gaussian filter is a low pass filter and the derivatives are band-pass filters. Differentiating the equation above with respect to ω_x and computing the extrema one, it is possible to determine the center frequency ω_0 of the nth (spatial) derivative.

$$\begin{aligned} \frac{d}{d\omega_x}\mathcal{G}_{x^{n_x}}\left(\omega_x, \sigma\right) &= \left(n_x - \omega_x^2\sigma^2\right)\jmath^{n_x}\omega_x^{n_x-1}e^{-\frac{\sigma^2\omega_x^2}{2}} = 0 \\ \omega_0 &= \pm\frac{\sqrt{n_x}}{\sigma} \end{aligned} \tag{3.8}$$

This equation states that the center frequency is coupled both with scale value σ and to the order of the derivative, this is illustrated in figure 3.3. As the order of the derivative increases, so does its center frequency and therefore, higher derivatives enhance higher level of spatial detail.

3.2.3 Filters based in Gaussian derivatives

Based in Gaussian derivatives up to the second order, it is possible to compute a set of invariant filters which are invariant to illumination variations and can be easily computed using the Half-Octave Gaussian Pyramid algorithm. This filter set has been used with success in texture, object and pixel classification (Griffin *et al.*, 2009; Lillholm and Griffin, 2008; Crosier and Griffin, 2008).

$$Mag(x,y,\sigma) = \sqrt{\left(G_x(x,y,\sigma)\right)^2 + \left(G_y(x,y,\sigma)\right)^2} \tag{3.9}$$

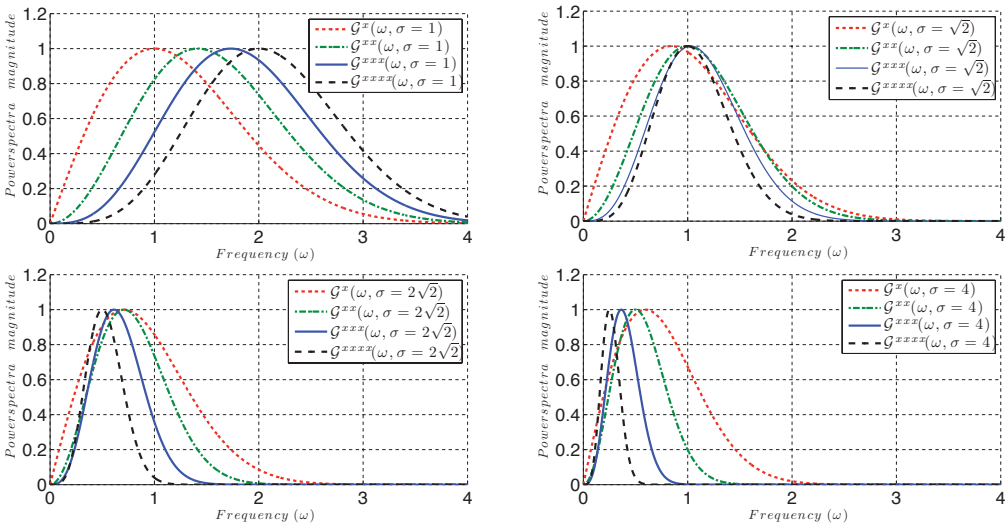


Figure 3.3: *Normalized power spectra for Gaussian derivatives up to fourth order computed at $\sigma = 1, 2, 4$ and 8. Gaussian derivatives act like band-pass filters*

$$LoG(x, y, \sigma) = \left(G_{x^2}(x, y, \sigma) + G_{y^2}(x, y, \sigma)\right) \tag{3.10}$$

$$\gamma(x, y, \sigma) = \sigma^2 \sqrt{\frac{1}{4}\left(G_x^2(x, y, \sigma) - G_y^2(x, y, \sigma)\right)^2 + G_{xy}(x, y, \sigma)^2} \tag{3.11}$$

where *Mag* corresponds to the gradient magnitude which is invariant to rotation changes and is computed with Gaussian derivatives of first order, *LoG* is the well known Laplacian of Gaussians operator, proposed by Lindeberg (1994) and γ is the third component of the second local order Gaussian jet norm, proposed by Griffin (2007). The main advantage of these filters is their invariance to strong changes in illumination present in facial analysis, specially in the face recognition application.

3.2.4 The Gaussian Jet

Koenderink and van Doom (1987) argue that the local visual appearance in an image neighborhood can be represented by a local Taylor series expansion of the neighborhood, computed using local Gaussian derivatives. The coefficients of this series constitute a feature vector, referred to as the "Local Jet" that compactly represents image appearance and can be used for indexing, matching and recognition. Romeny *et al.* (1993) have shown that invariance to scale and orientation can be obtained when the local jet is computed using Gaussian derivatives.

It be $I_{x^m y^n}(x, y, \sigma)$ an image filtered with a Gaussian derivative:

$$
\begin{aligned}
I_{x^m y^n}(x,y,\sigma) &= \left(\frac{\partial^{m+n}}{\partial^m x \partial^n y} G(x,y,\sigma)\right) * I(x,y) \\
&= \frac{\partial^{m+n}}{\partial^m x \partial^n y} \left(G(x,y,\sigma) * I(x,y)\right) \\
&= \frac{\partial^{m+n}}{\partial^m x \partial^n y} I(x,y,\sigma)
\end{aligned}
\tag{3.12}
$$

From the equation above, the Gaussian jet of kth order is defined as follows:

$$
\begin{aligned}
\vec{j}_k(x,y,\sigma) = \left(I(x,y,\sigma), I_x(x,y,\sigma), I_y(x,y,\sigma), ..., I_{x^m y^n}(x,y,\sigma)\right) \\
\vec{j}_k(x,y,\sigma) \in \Re^{\frac{(k+1)(k+2)}{2}} \quad \text{with} \quad m+n=k
\end{aligned}
\tag{3.13}
$$

The Second Local Order Gaussian Jet Norm

A mathematical definition of norm for a second order Gaussian jet ($k = 2$) has been proposed by Griffin (2007). This norm is defined as the minimum of scale space norms from a set of profiles measured by the jet at a given point in the image and is defined as:

$$
\left\|\vec{j}_2(x,y,\sigma)\right\| = \left(\sigma^2\left(I_x^2+I_y^2\right)+\tfrac{1}{4}\sigma^4\left(I_{x^2}^2+I_{y^2}^2\right)+\tfrac{1}{4}\sigma^4\left(\left(I_{x^2}-I_{y^2}\right)^2+4I_{xy}^2\right)\right)^{\frac{1}{2}}
\tag{3.14}
$$

The Second Order Gaussian Jet norm satisfies all the mathematical characteristics of a jet norm. In particular, it is unaffected by adding a constant to image intensities (Griffin, 2007) and thus can avoid problems due to changes in illumination .

3.3 The Half-Octave Gaussian Pyramid

Computing a scale invariant version of Gaussian derivatives for a $M = N \times N$ pixels image requires computing second order derivatives of the image at $log_2(N)$ scales. A linear complexity pyramid algorithm for this calculation has been known since the 1980's (Crowley and Parker, 1984). The result of this algorithm is a half-octave Gaussian Pyramid. An integer coefficient version of this algorithm (Crowley and Riff, 2003) has been demonstrated using repeated convolutions of the binomial kernel [1 2 1]. Implementations that compute such pyramids on real time exist for the current generation of computer stations, to demonstrate it, average time for the pyramid construction in several sizes of images is recorded and reported in table 3.1

The Gaussian pyramid for an $M = N \times N$ pixels image can be computed in $O(M)$ operations using cascade convolution with re-sampling (Crowley and Stern,

	128×128	256×256	512×512	1024×1024	320×240
Average construction time (ms)	0.507	2.181	9.072	36.852	2.233

Table 3.1: *Average (10.000 runs) pyramid construction time for five different input data sizes (in pixels), on an Intel®Pentium ®, 2.80 Ghz*

1984). This algorithm involves alternatively convolving with a Gaussian support, and re-sampling the resulting image with a sample distance of $\sqrt{2}$. The effect of cascade convolution is to sum the variances of the filters, so that the cumulative variance is $\sigma_k^2 = 2^k$ and the resulting standard deviation is $\sigma_k = 2^{\frac{k}{2}}$.

Interleaving re-sampling with convolutions decreases the number of image samples while expanding the distance between samples. This has the effect of dilating the Gaussian support without increasing the number of samples used for the Gaussian, effectively increasing the scale. Aliasing is avoided (or minimized) by the fact that the image has been low-pass filtered by previous convolutions. The result is an algorithm with linear algorithmic complexity (i.e. $O(M)$) producing a discrete representation of scale space with $2M$ total samples.

Let $P_k(x, y, \sigma)$ be the kth pyramid image. The input image $P(x, y, 0)$ is initially convolved with a filter of $\sigma_0 = 1$ to produce an initial image $P_0(x, y, 1)$.

$$k = 0 \rightarrow P_0(x, y, 1) = P(x, y, 0) * G(x, y, 1) \tag{3.15}$$

where "$*$" is the convolution operator. The pyramid image ($k = 1$) is produced by a convolution with the same low-pass filter, resulting in a cumulative scale factor of $\sigma_1 = \sqrt{2}$.

$$k = 1 \rightarrow P_1(x, y, \sqrt{2}) = P(x, y, 1) * G(x, y, 1) \tag{3.16}$$

Each successive image in the pyramid is computed by convolving an expanded Gaussian with a sampled image as described by the following recurrence equation:

$$P_k(x, y, \sqrt{2}^k) = \mathcal{S}_{\sqrt{2}^k}\left\{P_{k-1}(x, y, \sqrt{2}^{k-1})\right\} * \mathcal{E}_{\sqrt{2}^k}\{G(x, y, 1)\} \tag{3.17}$$

where $\mathcal{S}_{\sqrt{2}^k}\{\cdot\}$ is the "diagonal" resampling operator and $\mathcal{E}_{\sqrt{2}^k}\{\cdot\}$ is an diagonal expansion operator defined as follow:

$$\mathcal{S}_{\sqrt{2}^k}\left\{P_{k-1}(x, y, \sqrt{2}^{k-1})\right\} = \begin{cases} P_{k-1}(x, y, \sqrt{2}^{k-1}) & \text{if } (x+y)^2 \bmod (2^{k-1}) \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

$$\mathcal{E}_{\sqrt{2}^k}\{G(x,y,1)\} = \begin{cases} G\left(\frac{x+y}{2^{\frac{k}{2}}}, \frac{x-y}{2^{\frac{k}{2}}}, 1\right) & \text{if } (x+y)^2 \bmod (2^{k-1}) \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

The $k = 0$ image may be discarded or used for estimating a Laplacian image for $k = 1$ if required. Because the $k = 1$ image has been smoothed with a Gaussian low-pass filter of scale $\sigma = 2$, re-sampling with a sample distance of $\sqrt{2}$ will result in an aliasing of less 1% (Crowley and Riff, 2003).

Computing Gaussian Derivatives

Gaussian Derivatives are easily calculated in the x and y axis directions from the images in the pyramid by differences of adjacent pixels(for more details see appendix A).

3.4 Summary

This chapter summarizes Gaussian derivatives from a theoretical and practical scope. An overview of Gaussian scale space has been presented in section 3.1. In section 3.2, Gaussian derivatives were explained and their equations for steering them were presented. Moreover an analysis in frequency space was performed to justify the use of Gaussian derivatives up to the fourth order for face image detection and analysis. Furthermore a set of filters computed from the Gaussian derivatives up to second order and the Gaussian jet were presented.

In, Section 3.3, the Half-Octave Gaussian pyramid was introduced. Implementations of this pyramid make it possible to compute Gaussian derivatives at video rate and allow their use in real-world applications as face detection.

Détection de visages avec les Dérivés Gaussiennes

Le chapitre est consacre a l'explication de comme appliquer les dérivés de Gauss pour la détection de visages avec une cascades de classificateurs. Formellement, nous présentons une algorithme d'optimisation pour la cascade, basée dans le coût de calcul des dérivés de Gauss et l'information capture pour chaque dérivée. Dans ce processus, les dérivés dont son coût de calcul est réduit seront considérés dans les premiers nœuds de la cascade où la charge de calcul est plus fortement présente, la position des dérivés dans la cascade est choisi en tenant compte du taux de détection dans le nœud actuel.

Tous les concepts présentés dans ce chapitre validées par les expériences réalisées dans les bien connu base de données MIT-CMU et l'ensemble de données FDDB. Des expériences ont montré les avantages et les inconvénients des dérivés de Gauss dans la détection de visage. Outre les résultats de la charge de calcul et le taux de détection lors de transformations de l'image différente que la rotation, un bruit gaussien, le contraste et le flou sont également menées pour montrer les avantages des dérivés de Gauss dans le processus de détection de visage.

- Dans le MIT + visage CMU ensemble de données des résultats en utilisant des dérivés de Gauss ne sont pas supérieurs aux approches de l'art l'Etat dans la détection de visage, néanmoins cet ensemble de données est composé de faible qualité et des images numérisées qui affecte les performances de Gauss dérivés fonctionnalités pour détecter les visages . Résultats dans le visage FDDB ensemble de données montrent que les caractéristiques des dérivés gaussiens surpasser Haar-fonctions dans presque 4 % de différence dans le taux de détection. Nous tenons à souligner que FDDB est un visage difficile jeu de données avec les conditions du monde réel (voir les sections 4.5.2 et 4.6.2).

- Dans le test ensemble de données sensibles, gaussien caractéristiques des dérivés montrent une invariance à la rotation élevée et les variations de flou, mais l'invariance faible pour la présence de bruit dans les images, cela est dû à la présence d'ordres supérieurs dérivés (voir les sections 4.5.1 et 4.6.1).

- En charge de calcul, des cascades optimisée des fonctions gaussienne calculée dérivés de Pentecôte une pyramide demi-octave gaussien présentent une réduction de la charge de calcul (près de 30%) sur la version non optimisée en utilisant les fonctionnalités et les mêmes calculées Haar-whit caractéristiques de la images intégrales (voir les sections 4.5.3 et 4.6.3).

4

Face Detection with Gaussian Derivatives

Chapter Contents

4.1 Motivation **47**
4.2 Theoretical Background **48**
4.2.1 The cascade Architecture 48
4.2.2 Training a cascade of classifiers 49
4.2.3 Detecting faces with a cascade of classifiers 51
4.2.4 Computational Cost of cascade classifiers 51
4.2.5 Evaluation datasets 53
4.2.6 Cascade-Training Setup 56
4.3 Gaussian Derivatives as a feature set **56**
4.3.1 Experimental Protocols 57
4.4 Speed-optimized Cascades of Gaussian Derivatives **58**
4.5 Experimental Results with non-optimized cascades **61**
4.5.1 Sensitivity Results 61
4.5.2 Results on test data sets 61
4.5.3 Results in computational load 63
4.6 Experimental Results with speed-optimized cascades **65**
4.6.1 Sensitivity Results 65
4.6.2 Results on test data sets 66
4.6.3 Results on computational load 66
4.7 Discussion and Conclusion **67**

4.1 Motivation

A Practical method for real time face detection has been proposed by Viola and Jones (2001). Their approach is based on two ideas:

- The use of integral images to obtain an extremely large space of image features based on difference of boxes (Haar like features).
- The use of a cascade of linear classifiers learned using Adaboost.

The combination of these two techniques has led to the first real time face detector widely used in real world applications such as digital cameras and smartphones.

Unfortunately Haar features are sensitive to image plane orientation and low resolution images. This constraints limits their use in real world scenarios where the state-of-the-art face detectors based on those features are still not working properly (Jain and Learned-Miller, 2010; Zhang and Zhang, 2010).

To address this problems we propose to use a cascade of Gaussian derivative features up to the fourth order computed in real time with a half octave Gaussian pyramid. In addition, we propose a speed-cascade optimization based in the computational cost of Gaussian derivatives and the captured information necessary to perform an adequate detection process.

The chapter is organized as follows: a theoretical background about cascade of classifiers and its training is exposed in section 4.2; in section 4.3, Gaussian derivatives are presented as a feature set for training cascades of classifiers. A cascade framework for training speed-optimized cascades of Gaussian derivatives features is presented in section 4.4 and experimental results are presented in sections 4.5 and 4.6. Finally section 4.7 closes the chapter with some concluding remarks.

4.2 Theoretical Background

4.2.1 The cascade Architecture

A cascade of classifiers is organized as a cascade of classification stages. The algorithm hypothesizes the existence of a face at a particular reference position (x, y) and scale s in the image. The reference position is used to specify a set of image features that are sent to the cascade of linear classifiers.

The cascade of linear classifiers is composed of a number of stages. Each stage combines the vote of a small set of weak classifiers $\mathbf{h} = (h_1, h_2, \ldots, h_T)$ in a strong classifier $H(\mathbf{z})$:

$$H(\mathbf{z}) = \begin{cases} +1 & if \quad \sum_{i=1}^{T} a_i h_i(\mathbf{z}) \geq b \\ -1 & \text{otherwise} \end{cases} \tag{4.1}$$

with:

$$h_i(\mathbf{z}) = \begin{cases} +1 & if \quad p_j f_j(\mathbf{z}) < p_j \tau \\ -1 & \text{otherwise} \end{cases} \tag{4.2}$$

Where a_i are the node weights, b is the node's threshold, p_j is the parity coefficient (-1 or 1) which indicates the inequality's direction and τ is the weak-classifier's threshold. Each weak classifier provides a yes/no decision based on a single image feature $f_j(\mathbf{z})$ evaluated relative to the reference position and scale. The committee of weak classifiers at each stage can vote to reject the hypothesis at the reference position and scale, or to pass the hypothesis to a next more computational expensive stage.This procedure is repeated to provide a cascade of classifiers that increasingly concentrate to reduce the number of difficult sub-windows as shown in figure 4.1.

From this description,the final detection rate of a cascade of classifiers D is computed as the product of detection rates in each node, the same is valid for the final false positive rate F as shown in the next equation:

$$D = \prod_{i=1}^{r} d_i \quad F = \prod_{i=1}^{r} false_i \tag{4.3}$$

Where d_i corresponds to the detection rate in the node i and $false_i$ corresponds to the false positive rate in the node i.

4.2.2 Training a cascade of classifiers

Training a cascade of classifiers requires solving an optimization problem where each stage must detect almost all the positive instances (faces) and reject a considerable percentage of negative ones (non-faces). The first layer has a small number of weak classifiers that reject a pre-defined percentage of negative examples and detect nearly 100% of the positive ones in the training dataset. The next layer is then trained to reject the same percentage of negative examples and detect nearly of 100% of positive examples using the detected false positives as a result to apply the current cascade in a bootstrapping image dataset. This process is summarized by Wu *et al.* (2008) in the algorithm 1

The Node-learning step in the algorithm 1 is composed by two algorithms used in in the experimental section to train the cascades :

- Adaboost (Freund and Schapire, 1997) to find out the best weak classifiers from a high dimensional feature set.

  At each iteration, Adaboost procedure gets a new weak classifier taking into account the classification error in a weighted distribution of the training set, such distribution is updated at each iteration giving more importance at the set of training examples misclassified by the precedent weak classifier (higher weights are giving to these examples) thus focusing on the examples that are hard to classify.

  The principal advantage of Adaboost is that the training error converges exponentially towards zero and the generalization performance grows at each iteration when the null training error is reached by the algorithm.

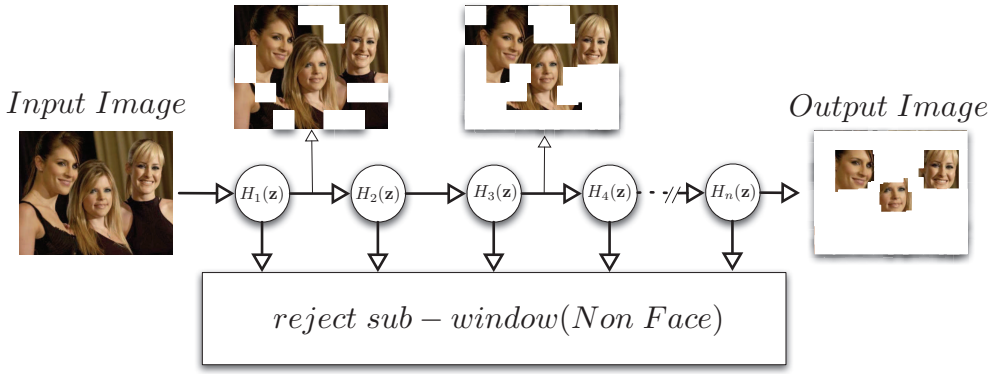


Figure 4.1: *Overview of the cascade architecture. At each stage, the classifier either rejects the sample and the process stops, or accepts it and the sample is forwarded to the next stage*

Algorithm 1 The cascade framework (Wu *et al.*, 2008)

{Giving a set of positive examples $\mathcal{P}$, a set of initial negative examples $\mathcal{N}$ and a set of bootstrapping negative examples $\mathcal{D}$.}.
{Giving a training learn goal G}.
{The output is a cascade $H = (H_1, H_2, H_3, \ldots, H_n)$}
$i \leftarrow 0, H \leftarrow 0$
repeat
  $i \leftarrow i + 1$
  Node Learning { Learn H_i using $\mathcal{P}$ and $\mathcal{N}$, add H_i to H}
  Remove correctly classified non-face patches from $\mathcal{N}$
  Run the current cascade H on $\mathcal{D}$, add any false detection to $\mathcal{N}$ until $\mathcal{N}$ reaches the same size as the initial set.
until The learning goal $\mathcal{G}$ is satisfied

- LAC (Linear Asymmetric Classifier) (Wu *et al.*, 2008) that guarantee an optimal linear strong classifier (see equation 4.1) to accomplish the node-learning goal, keeping the best trade-off between performance and computational cost. LAC is the result of solving the next asymmetric node learning optimization:

$$\begin{aligned} \max_{a \neq 0, b} \quad & \Pr_{\mathbf{x} \sim (\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})} \left\{ \mathbf{a}^T \mathbf{x} \geq b \right\} \\ \text{s.t.} \quad & \Pr_{\mathbf{y} \sim (\bar{\mathbf{y}}, \Sigma_{\mathbf{y}})} \left\{ \mathbf{a}^T \mathbf{y} \leq b \right\} = \beta \end{aligned} \tag{4.4}$$

Assuming a normal distribution ($\beta = 0.5$) in $\mathbf{a}^T\mathbf{y}$ term, the solution for the above equation is:

$$
\begin{aligned}
\bar{\mathbf{x}} &= \frac{\sum_{i=1}^{n_{\mathbf{x}}} \mathbf{h}(\mathbf{x}_i)}{n_{\mathbf{x}}}, \quad \bar{\mathbf{y}} = \frac{\sum_{i=1}^{n_{\mathbf{y}}} \mathbf{h}(\mathbf{y}_i)}{n_{\mathbf{y}}} \\
\Sigma_{\mathbf{x}} &= \frac{\sum_{i=1}^{n_{\mathbf{x}}} \left(\mathbf{h}(\mathbf{x}_i) - \bar{\mathbf{x}}\right)\left(\mathbf{h}(\mathbf{x}_i) - \bar{\mathbf{x}}\right)^T}{n_{\mathbf{x}}} \\
\Sigma_{\mathbf{y}} &= \frac{\sum_{i=1}^{n_{\mathbf{y}}} \left(\mathbf{h}(\mathbf{y}_i) - \bar{\mathbf{y}}\right)\left(\mathbf{h}(\mathbf{y}_i) - \bar{\mathbf{y}}\right)^T}{n_{\mathbf{y}}} \\
\mathbf{a} &= \Sigma_{\mathbf{x}}^{-1}\left(\bar{\mathbf{x}} - \bar{\mathbf{y}}\right), \quad b = \mathbf{a}^T \bar{\mathbf{y}}
\end{aligned}
\tag{4.5}
$$

Where n_x is the number of positive examples (faces), n_y is the number of negative examples (non-faces), $\mathbf{x_i}$ are the positive examples (faces) and $\mathbf{y_i}$ the negative ones (non-faces)

4.2.3 Detecting faces with a cascade of classifiers

To detect faces with a cascade of classifiers, the most well known method is to use a sliding window which is moved using a step size $\Delta_{x,y}$ across multiples scales and locations. The size of the sliding window or the size of the features in the cascade can be modified by a factor s (usually fixed between 0.7071 and $\sqrt{2}$) in order to detect faces at different sizes. In each visited location, the cascade is applied to verify the presence of a face.

4.2.4 Computational Cost of cascade classifiers

A first measure to provide quantitative evaluation of run-time computational cost was proposed by Viola and Jones (2004). This measure computes computational load as the expected number of applied features. If the nodes $i = 1, \ldots, N$ pass fractions p_i of the search windows to the next node at a cost of M_i features for the node i, then the expected computational cost or load T is giving by:

$$
E[T] = \sum_{i=1}^{N} M_i \prod_{j=1}^{i-1} p_j \tag{4.6}
$$

Later, Brubaker *et al.* (2008) proposed an extension of this equation which computes the entire computational cost of deciding a sub-window for the node, including the cost of features belonging to previous nodes.

$$
E[T_i] = r_i \sum_{k=1}^{i} M_k \tag{4.7}
$$

Algorithm 2 Detecting faces with a cascade of classifiers using a sliding window

{The input is an image I with dimensions (I_{width}, I_{height}) }.
{Giving a cascade of classifiers $\mathcal{H}$ trained with a feature set $\mathcal{F}$ }.
{Giving a scale factor s and an initial size of sliding window (w_{width}, w_{height})}.
{Giving an step size $\Delta_{x,y}$}
{The output is a list $\mathcal{L}$ containing the positions (x_d, y_d) and sizes of detected faces}
$i \leftarrow 0,\ \mathcal{L} \leftarrow \emptyset,\ (w_{tempWidth}, w_{tempHeight}) \leftarrow (w_{width}, w_{height})$
compute the image representation I_R of the input image I
repeat
  Run the cascade H on I_R with a step size $\Delta_{x,y}$ and record all the detect positions in a temporal list $\mathcal{L}$.
  if $s < 1$ **then**
    $i \leftarrow i + 1$
    resize I by a scale factor s^i
    $I_{tempWidth} \leftarrow I_{width} * s^i$
    $I_{tempHeight} \leftarrow I_{height} * s^i$
    compute the image representation I_R of the resized image I
  else
    $i \leftarrow i + 1$
    resize the features in the cascade H by a scale factor s^i
    $w_{tempWidth} \leftarrow w_{width} * s^i$
    $w_{tempHeight} \leftarrow w_{height} * s^i$
    $\Delta_{x,y} \leftarrow [s\Delta_{x,y}]$
until {
if $s < 1$ **then**
  $(I_{tempWidth} < w_{width}) \wedge (I_{tempHeight} < w_{height})$
else
  $(w_{tempWidth} > I_{width}) \wedge (w_{tempHeight} > I_{height})$
}
merging similar detections in the list $\mathcal{L}$ to obtain a single detection per face

where:

$$r_i = (1 - p_i) \prod_{j=1}^{i-1} p_j \tag{4.8}$$

Where $E[T_i]$ is the expected computational load for a stage i in the cascade, M_k is the number of features in the node k and r_i is the fraction of the *decided sub-windows* in the node i.

Brubaker *et al.* (2008) defined the *decided sub-windows* in a node i as the sub-windows that it does not pass on, either by rejecting the sub-window as a non-face, or by accepting the instance as a face, if the node is the terminal one.

As shown in equation 4.7 the computational load is calculated based on the

number of features applied by a layer. To extend more this concept and compare cascades with different types of features, we propose to use the number of requests per layer R_k.

$$E[T_i] = r_i \sum_{k=1}^{i} R_k \tag{4.9}$$

A *request* is defined as a simple memory access made by the cascade to the image representation (e.g. integral image, Half-Octave pyramid), multiple requests may be necessary to compute a simple feature in a k node of the cascade.

For example, for computing I_x (see apendix A) in a specific position and scale (x, y, σ), three requests to the half-octave Gaussian pyramid are necessary. In the case of higher order derivatives such as I_{x^3} five requests are required. The examples before mentioned are related only with σ available values in integer levels of the pyramid, in the case of non-integers values, more requests could be necessary. For more details about the number of requests in the half-octave pyramid and the integral image, please see the Apendix A.

Computational load can be also expressed in a single image as a function of image location (Fleuret and Geman, 2002). Starting from a black background, the pixel intensities are modified proportionally to the number of requests applied to that window. For smallest sub-windows, intensity is concentrated in the central pixel of the window; for larger sub-windows, intensity is extend over a central region with an area proportional to the sub-window analyzed.

4.2.5 Evaluation datasets

Two public available face datasets were used in our experiments to compute all our ROC curves: MIT+ CMU and FDDB face datasets.

Sensitivity testing dataset

To analyse the performance of a cascade of classifiers when the input images are under the effects of some transformations commonly present in real-world images, we conducted an experiment using a sensitivity testing dataset. To do this, we select a set of 20 images (see figure 4.2a) from different subjects present in the Labelled Faces in the Wild dataset(LFW) (Wolf *et al.*, 2009; Huang *et al.*, 2007). All the used images were normalized to a size of 250 × 250 pixels. Notice than we use the LFW images proposed by Wolf *et al.* (2009) in which the faces were rectified in orientation and position with commercial face alignment software. Finally a set of image-transformations are applied in the normalized images, the applied transformations are:

- **Rotation** Each image is rotated sequentially by an angle varying between -25 and +25 degrees with a step of 3 degrees (see figure 4.2b).

- **Blurring** : A Gaussian smoothing filter with scales ranging from 0 to 10 is applied to each image (see figure 4.2c).
- **Noise**: Gaussian white noise with mean 0 and standard deviation between 0 and 0.2 is added to each image(see figure 4.2d).
- **Contrast**: For each image, the pixel intensities I_p are modified as stated by $I_p = \alpha I_m + (1 - \alpha) I_p$, where I_m is the mean intensity of the image and α is a parameter varying from -2 to 1.0. (see figure 4.2e).

CMU + MIT dataset

The MIT+CMU face dataset was introduced by Rowley *et al.* (1998) for testing, Some image examples of this dataset are shown in figure 4.3.

The first version of this test set contained 23 images with a total of 155 very low-resolution faces. The complete set contains 130 images with 507 faces. However, some of these annotated faces are manually drawn and these are counted as false detections in some publications.

In our experiences we used the complete face dataset and a detected face is considered as valid if the face sub-window contains all the fiducial points given in the ground-truth and its size is not twice larger than the minimum square that contains all the fiducial points.

The MIT+CMU dataset is one of the most widely used face datasets. Nevertheless, images contain artifacts that are not characteristic of those found in most real-world applications. Thus, we need to conduct experiments with a realistic dataset, such a dataset must contain images taken under real-world conditions presented for example in a surveillance system or a normal multimedia device as a mobile-phone cam or a digital camera. To deal with this problem we conduct more experiments in a new challenging face dataset with a much larger number of faces and more accurate annotations for the face regions than the MIT+CMU dataset.

Face detection and dataset benchmark (FDDB)

The FDDB (Jain and Learned-Miller, 2010) is a new challenging data-set of face images with a much larger number of faces and more accurate annotations for the face regions than previous datasets. Some examples of this dataset are shown in figure 4.4, the images in this dataset contain large variations in pose, lighting, background and appearance.

The FDDB dataset contains:

- 2845 images with a total of 5171 faces.
- A wide range of difficulties includes occlusions, difficult poses, and low resolution and out-of-focus faces.

(a) *Original sensitive dataset*

(b) *Rotation*

(c) *Blurring*

(d) *Gaussian White Noise*

(e) *Contrast*

Figure 4.2: *Example images from the sensitive test dataset (images modified from LFW (Wolf* et al.*, 2009; Huang* et al.*, 2007))*

Figure 4.3: *Example images from the MIT + CMU face dataset*

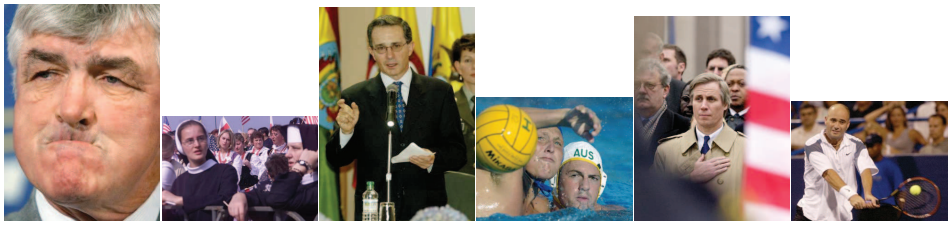


Figure 4.4: *Example images from the FDDB face dataset*

- An effective specification of face regions as elliptical regions.

Two different experimental protocols have been used:

- 10 fold cross-validation: A 10 fold cross-validation is performed using a fixed partitioning of the data set into ten folds. The cumulative performance is reported as the average curve of the ten ROC curve computed in each fold.
- Unrestricted Training: Data outside the FDDB data set is permitted to be included in the training set. The ten folds are separately used as validation sets to obtain ten different ROC curves. The cumulative results are reported as mentioned for the 10 fold cross-validation.

The ten-fold testing as well as the implementation-software of the algorithms for matching detections and annotations are publicly available[1]. From this software, a detection is scored taking into account the next equation:

$$S(d_i, l_j) = \frac{area(l_j) \cap area(d_i)}{area(l_j) \cup area(d_i)} \tag{4.10}$$

Two different types of ROC curves could be computed using the above mentioned score. The first one is the discrete score curve, where only the detection scores superior to 0.5 are used an the second is the continuous score curves where all the possible detections scores are included.

4.2.6 Cascade-Training Setup

To train all the cascades employed in this thesis, we use a training set containing 5000 example face images and 5000 initial non-face examples, all of size 24×24 pixels. A set of 4832 face images are used for validation purposes. We used approximately 2,284 million non-face patches to bootstrap the non-face examples between nodes. We require that every node have 50 percent false positive rates and the cascade training process is terminated when there are not enough non-face patches to bootstrap. In order to make the face detector run at video speed, the first node uses only seven features, we use more features as the node index increases (the last node used 200 features).

4.3 Gaussian Derivatives as a feature set

The choice of feature set has important impact on detection rate as well as the scan speed in the final cascade. We have explored a feature space composed by derivative orders up to four. Derivatives are computed at four different orientations $\theta = \{0, \pi/4, \pi/2, 3\pi/4\}$ in a 24×24 pixel window for all the real

[1] http://vis-www.cs.umass.edu/fddb

sample positions available in a Gaussian pyramid of three levels $\sigma = \left\{\sqrt{2}, 2, 2\sqrt{2}\right\}$. The final set of Gaussian derivatives features available in our experiments is 8064 derivatives.

To test the performance of Gaussian derivatives features, we define four different feature sets as show in Table 4.1. We train four cascades (one for each feature set) using the algorithms and the training sets explained in preceding sections. Each cascade has 21 nodes, except for the cascade trained with the feature set number 3 that has 22. From each trained cascade, we measure the node performance as the false negative rate in the validation set for each node in each cascade and we show the results in the Figure 4.5. The experiment demonstrates that adding high-order Gaussian derivatives improves performance. In this case, detection outperforms higher orders in the first three nodes and then for deeper nodes the Gaussian derivatives up to fourth order dramatically improve the node-performances.

The node performance measure is useful because it directly compares the ability of each feature set to achieve the node-learning goal with a small number of features per node and number of nodes in the cascade.

4.3.1 Experimental Protocols

We conducted several experiments to demonstrate the performance of Gaussian derivatives compared to Haar-features in the face detection problem, the experiments are divided as follows:

- **Sensitivity analysis**: we made a sensitive testing dataset, where transformations are applied in the images for conducting experiments to evaluate their influence in the cascade performance (detection rate), We applied all the cascades in our experiments to each image in the transformed dataset. For the transformation parameters listed in section 4.2.5, we record the detection rate over the set of images and the number of eventual false positives. The results are reported in sections 4.5.1 and 4.6.1.

- **Comparative results in test datasets**: The performance of the cascade is commonly measured by a ROC (Receptive Operator Curve) calculated with an evaluation dataset. In all our experiments we resize the sliding window

Feature Set	Derivative Orders	Total
1	Only First Order	1792
2	First + Second Orders	4032
3	First + Second + Third Orders	5824
4	All Available Orders	8064

Table 4.1: *Four different feature sets using different Gaussian derivative orders at pyramid levels of* $\sigma = \left\{\sqrt{2}, 2, 2\sqrt{2}\right\}$ *and orientations* $\theta = \{0, \pi/4, \pi/2, 3\pi/4\}$

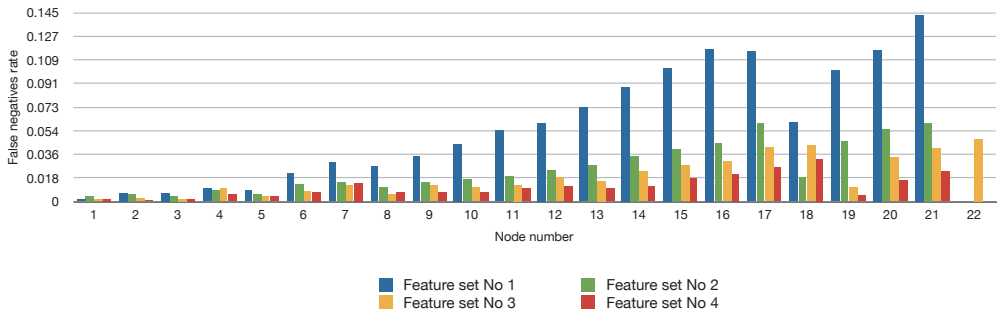


Figure 4.5: *Node performances for the four cascades trained with the feature sets show in the table 4.1. The node error decreases when the derivative order rises, specially for deeper nodes in the cascade. Notice also that the cascade trained using the feature set 4 has only 21 nodes which show a high performance of the features to reach the fixed node detection and positive rates*

by a factor $s = 1.20$ which is a common used value in face detection benchmarks. Finally, we compare all our results with a cascade of Haar-like features to show the performance of our approach compared with the state-of-the-art methods (details about Haar-features in appendix A). The results are reported in section 4.5.2 and 4.6.2.

- **Computational load**: For all our cascades, we analyze the computational load in the FDDB face dataset, using an scale factor $s = 0.833$. The computational load is calculated as is explained in section 4.2.4 and we report the results in sections 4.5.3 and 4.6.3.

All the cascades tested in this thesis were trained with the same conditions and training set parameters described above and all the evaluations are performed with strictly identical parameters.

4.4 Speed-optimized Cascades of Gaussian Derivatives

In the preceding section, we have observed the effects of adding Gaussian derivative features up to fourth order in the cascade framework. We have observed that a strong improvement is obtained in the deeper nodes of the cascade. At the same time, higher order derivatives have a slightly higher computational cost. Thus for performance reason, it is better that they be used only in deeper levels of the cascade.

From this assumption, many researchers have proposed to combine different feature types in the same cascade to improve the detection speed. Meynet *et al.* (2007) proposed use in the first five nodes Haar-like features and in the final nodes they used anisotropic Gaussian filters. Xiaohua *et al.* (2009) use Haar-like features in the first nodes and then an approximation of Gabor filters computed from an integral image.

Despite the detection speed improvement different image representations must be computed and this could be a trade-off not desired in embedded systems due

to memory space retained for each of they during the detection process in a single image. We have explored the use of a new optimized cascade framework that uses always the same image representation, in our case the scale space representation computed by the Gaussian pyramid.

Our cascade framework takes advantage from the information captured for each derivative in order to keep the performance of the cascade. Experiments performed have showed that Gaussian derivative features of first order retain edge information, second orders retain blob shapes and superior orders retain detailed information that could be used to distinguish a face.

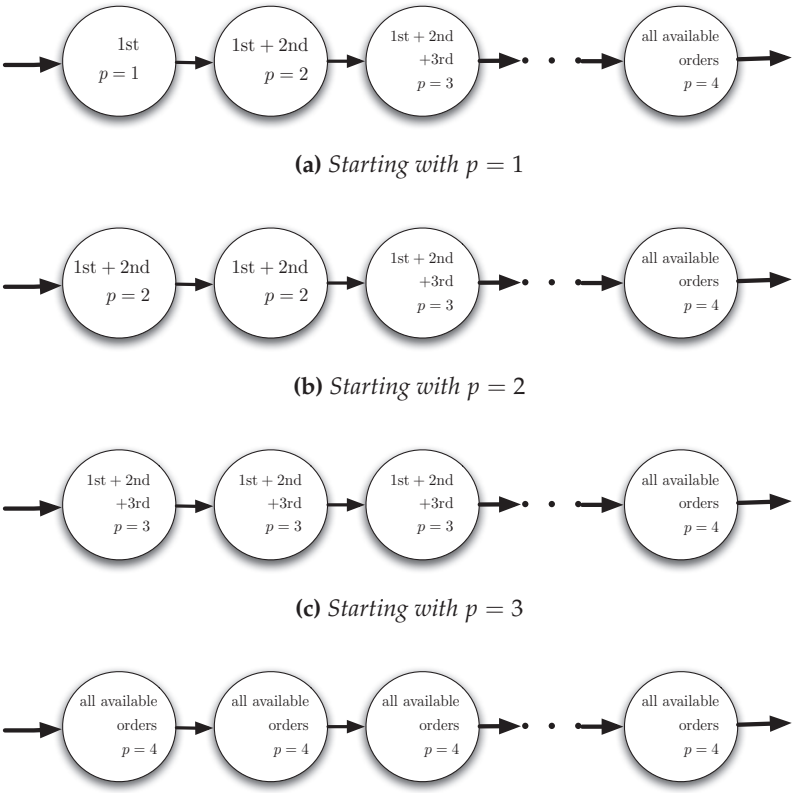
From the preceding premise, we can expect that Gaussian derivatives features of lower order are going to perform well in the first nodes when not much information is necessary to discriminate a face and features of lower computational cost are expected due to high quantity of windows to analyze. In other way higher orders could be useful in only deeper nodes where the difference between a face and a background image becomes more difficult and the quantity of windows is minimal to apply computational expensive features. A graphical example is shown in figure 4.6.

To deal with this, we propose a new Optimized cascade framework that uses the mathematical and computational properties of Gaussian derivatives features to improve detection speed without considerable lost of performance. The algorithm 3 summarizes our approach to train speed-optimized cascades.

Algorithm 3 The speed-optimized cascade framework

{Giving a set of positive examples $\mathcal{P}$, a set of initial negative examples $\mathcal{N}$, a set of positive validation examples $\mathcal{V}$ and a set of bootstrapping negative examples $\mathcal{D}$.}.
{Giving a training learn goal G}.
{Giving a training learn goal per layer G_L}.
{Giving an ensemble of p feature sets $\mathcal{F} = (F_1, F_2, F_3....F_p)$}.
{The output is a cascade $H = (H_1, H_2, H_3, ..., H_n)$ }
$i \leftarrow 0$,$H \leftarrow 0$,$p \leftarrow 1$
repeat
  $i \leftarrow i + 1$
  Node Learning { Learn H_i using $\mathcal{P}$, $\mathcal{F}_p$ and $\mathcal{N}$, add H_i to H}
  Run the current node H_i on $\mathcal{V}$ to compute d_i
  while $d_i < G_L$ **do**
    $p \leftarrow p + 1$
    Node Learning { Learn H_i using $\mathcal{P}$, $\mathcal{F}_p$ and $\mathcal{N}$, add H_i to H}
  Remove correctly classified non-face patches from $\mathcal{N}$
  Run the current cascade H on $\mathcal{D}$, add any false detection to $\mathcal{N}$ until $\mathcal{N}$ reaches the same size as the initial set.
until The learning goal $\mathcal{G}$ is satisfied

To test the viability of our approach for keeping the detection performance, we train four different cascades fixing the parameter p from the algorithm 3 to $p =$



(a) *Starting with* $p = 1$

(b) *Starting with* $p = 2$

(c) *Starting with* $p = 3$

(d) *Non-optimized cascade*

Figure 4.6: *Graphical example of our speed-optimized cascade framework for different values of p. The distribution of features in each node of the cascade takes in consideration the computational cost of each derivative order*

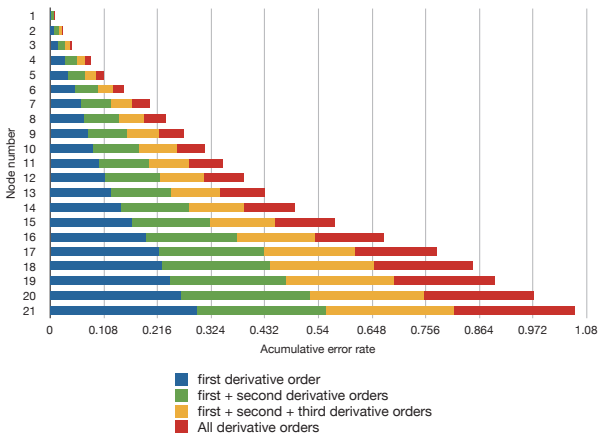


Figure 4.7: *Accumulative error rate across the cascade nodes using* $p = \{1, 2, 3, 4\}$ *in the speed-optimized cascade training framework. Adding higher derivative order decreases the error rate, specially after the node number 5.*

$\{1, 2, 3, 4\}$, then we compute the accumulative error rate across the cascade nodes and we report the results in Figure 4.7.

4.5 Experimental Results with non-optimized cascades

This section presents all the comparative experimental evaluation using a non speed-optimized cascade of Gaussian derivative features and a cascade of Haar features, both of them trained using Adaboost + LAC.

4.5.1 Sensitivity Results

The results in the sensitivity test data set are shown in figure 4.8.

The sensitivity of rotation can be observed in figure 4.8a, in this case Gaussian derivatives outperform Haar features cascades. In effect, the detection rate for cascades of Gaussian derivatives is 100% for angles between -13 and +5 degrees and decreases significantly for larger rotations. On the other hand Haar features are very sensitive to rotation variations with only a detection rate of 100% for angles between -3 and +3 degrees. The number of false positives for this experiment was zero in all the cases.

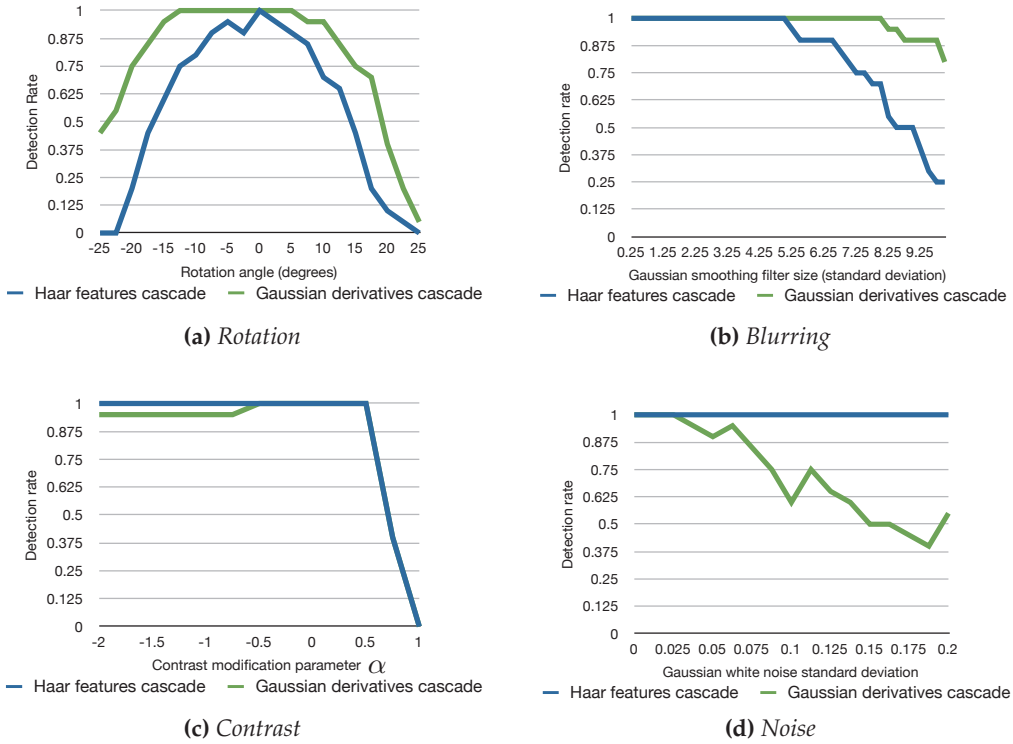
The results of the influence of blurring are reported in figure 4.8b. One can observe that the detection rate if still 100% for an standard deviation of 8.3 and decreases slowly after. Compared with a cascade of Haar features which has still a detection rate of 100% for an standard deviation of 5 and then decreases quickly. Any false positive was noticed in this experiment.

Tolerance to contrast is reported in figure 4.8c, as we can see cascades of Gaussian derivatives has competitive results compared with the Haar-features cascade. Notice that in our cascade no intent to normalize illumination was performed. Experiments using illumination normalization showed no noticeable improvements.

The influence to noise is reported in figure 4.8d. Despite the robustness to noise of Gaussian derivatives cascade, the cascade of Haar features outperform with 100% of detection rate (any false positive) for all the standard deviation values used in the Gaussian noise, compared with the cascade of Gaussian derivatives that has 100% of detection rate for values lower than 0.028 (one false positive for a value of 0.06) and decreases slowly as the noise increases. This results are due to the variability to noise of higher order Gaussian derivatives (up to fourth order), this problem was explained in section 3.2.2,

4.5.2 Results on test data sets

In order to compare our approach with other state-of-the-art methods, we consider reported results in the MIT+CMU face data and we show the results in figure 4.9 and table 4.2. As we can see, Gaussian derivatives do not achieve a very good



(a) *Rotation* **(b)** *Blurring*

(c) *Contrast* **(d)** *Noise*

Figure 4.8: *Results of comparing a non-optimized cascade of Gaussian derivatives with a cascade of Haar features in the sensitivity testing dataset. In this experiment Gaussian derivatives showed a high invariance to rotation angle (a) and blurring (b) in the other hand a high invariance to image noise is exhibit by Haar features*

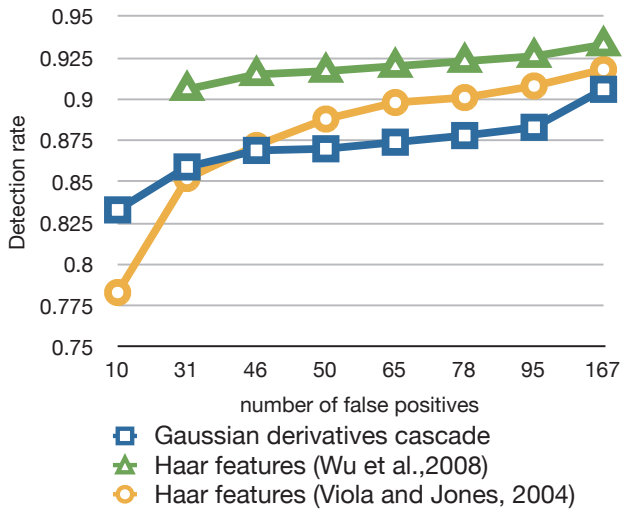


Figure 4.9: *Performance comparison with previous works in the CMU+MIT face dataset*

performance compared with the state-of-the-art face detectors. In this scope, we obtain almost 6 % of detection rate difference with a low number of false positives and 4% of difference with a bigger number of false positives.

This results confirm the sensitivity to noise when Gaussian derivatives of higher order are used. We can observe the following faults in the testing data set as:

- Our cascade was trained to deal with a high resolution facial images that can be found in recent multimedia systems as mobile phones and digital cameras. The MIT+CMU face data set is composed images that were scanned from newspapers and are not a representative example of images obtained from modern digital cameras.
- Aliasing in some images of the data set due to a low-quality scanning process. In effect, Aliasing is rarely seen with digital cameras, because digital cameras almost always use intentional blurring in front of the CCD to avoid aliasing. In our case artefacts caused by aliasing are increased due to high-frequencies presence.

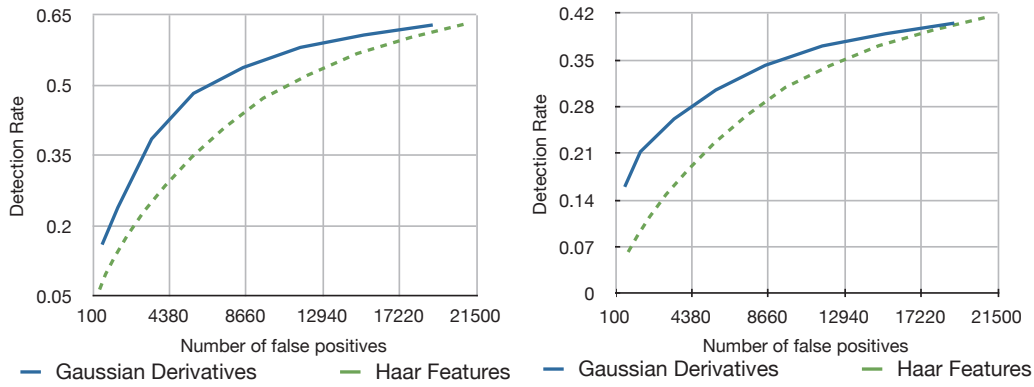
To confirm the problems founded in the MIT+CMU face data set, we have tested our cascade using the FDDB face dataset and we show the results In figure 4.10. For this data set the evaluation was performed using the discrete and continuous scores as is shown in figures 4.10a and 4.10b respectively. In this case, the cascade of Gaussian derivative features outperform the cascade of Haar features with almost a 8% of difference in the detection rate (area under the ROC). Results in this data set confirms the high-precision and selectivity of Gaussian derivatives features to locate a face in an image.

4.5.3 Results in computational load

Results comparing the computational load of a non-optimized cascade of Gaussian derivatives with a cascade of Haar features are reported in figure 4.11. Despite the

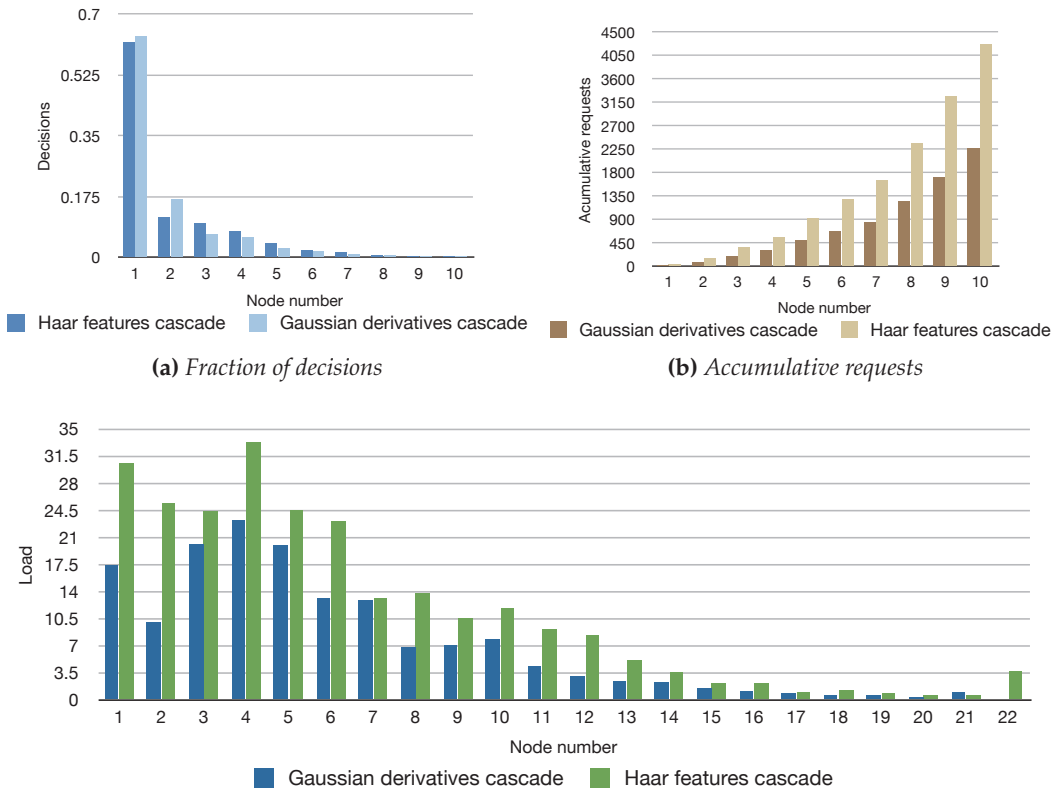
	False Positives								
Method	**6**	**10**	**31**	**46**	**50**	**65**	**78**	**95**	**167**
Viola and Jones (2004)	-	0.783	0.852	-	0.888	0.898	0.901	0.908	0.918
Garcia and Delakis (2004)	-	0.905	0.915	-	-	0.923	-	-	0.931
Osadchy *et al.* (2007)	-	-	-	-	-	0.830	-	-	-
Li and Zhang (2004)	-	0.836	0.902	-	-	-	-	-	-
Luo (2005)	0.866	0.874	0.903	-	0.911	-	-	-	-
Schneiderman (2004)	0.897	-	-	0.957	-	-	-	-	-
Rowley *et al.* (1998)	-	0.832	0.86	-	-	-	-	-	0.901
Haar features (Wu *et al.*, 2008)	-	-	**0.906**	**0.915**	**0.917**	**0.920**	**0.923**	**0.926**	**0.933**
Gaussian Derivatives	-	**0.833**	**0.859**	**0.869**	**0.870**	**0.874**	**0.878**	**0.883**	**0.906**

Table 4.2: *A comparison of detection rates on the CMU+MIT data set for several standard detectors*



(a) *Discrete Score* **(b)** *Continuous Score*

Figure 4.10: *Performance comparison of a non-optimized cascades of Gaussian derivatives features with a Haar-features cascade in the FDDB face dataset. As we can see, the non-optimized cascade of Gaussian derivatives outperform in this dataset.*



(a) *Fraction of decisions* **(b)** *Accumulative requests*

(c) *Estimated Load*

Figure 4.11: *Computational Load comparison between a cascade of Haar features and a non-optimized cascade of Gaussian derivatives. The number of accumulative requests (b) for a cascade with Gaussian derivatives is inferior to a cascade with Haar features, as consequence the estimated load (c) in a cascade with Gaussian is reduced, specially in the first nodes.*

fast computation of Haar features using integral image and the superior number of decisions taken in earlier nodes, the cascade of Gaussian derivatives, requires fewer requests to the pyramid to accomplish a better task in the FDDB face dataset (see figure 4.11b); as a consequence the computational load required for the cascade of Gaussian derivatives is less than the Haar-features cascade in the nodes in the cascade (see figure 4.11c). Also notice than the cascade of Gaussian derivative features has only 21 layers compared to 22 in the Haar features cascade which is an advantage in memory requirements for storing the cascade in possible embedded systems where the amount of available memory is limited.

In figure 4.12, we report an example of computational load as a function of image position, notice than the sub-window positions with a level of intensity higher have required more requests and as consequence more computational load to be discriminated by the cascade.

4.6 Experimental Results with speed-optimized cascades

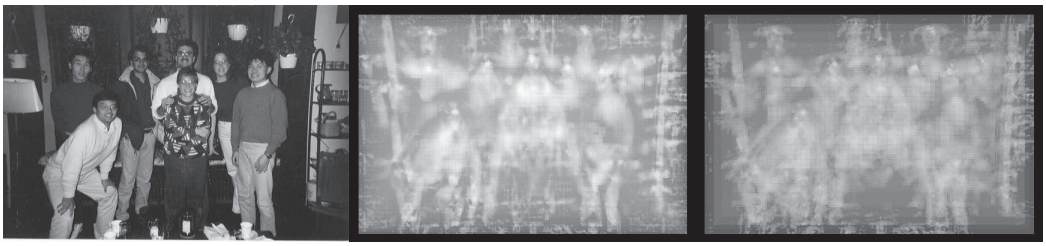
This section presents all the comparative experimental evaluation using a speed-optimized cascade framework.

4.6.1 Sensitivity Results

In figure 4.13, we report the comparative results in the sensitivity testing data set using different values of p.

The results of sensitivity to rotation are shown in figure 4.13a. In this experiment, the optimized cascades continues to operate in the same range of orientations as the non-optimized version with a range of -12 to +8 degrees (detection rate of 100%). Clearly the speed-optimized cascades are not influenced by the variations in rotation.

The sensitivity of blurring using the speed-optimized cascade framework can be observed in figure 4.13b, notice than cascades trained using $p = 2$ have a



(a) *Original Image* **(b)** *Cascade of Haar-features* **(c)** *Non-optimized cascade*

Figure 4.12: *Computational load shown as function of image location: a) Image Original, b)with a cascade of Haar features and c) with a non-optimized cascade of Gaussian derivatives. notice than the sub-window positions with a level of intensity higher have a high computational load. In the case of Gaussian derivatives features the computational load is reduced as we can see by its intensity levels (c)*

better performance with almost a 100% of detection rate for standard deviations lowers than 9 (any false positive was reported for the optimized cascades in this experiment).

Analysis to contrast variations are reported in figure 4.13c, once again, the speed-optimized cascade trained using a value of p, outperform with a detection rate of 100% for values of α lower than 0.5. Any false positive was perceived in this experiment.

Finally, the influence to noise is reported in figure 4.13d, as we can see the cascade trained using $p = 2$ outperform the rest of optimized cascades with a detection rate of 100% for standard deviations of Gaussian noise lower than 0.05. In this experiment, one false positive was detected at 0.06 for the optimized cascades with values of $p = 1$ and 2 and other one was detected at 0.05 with a value of $p = 3$.

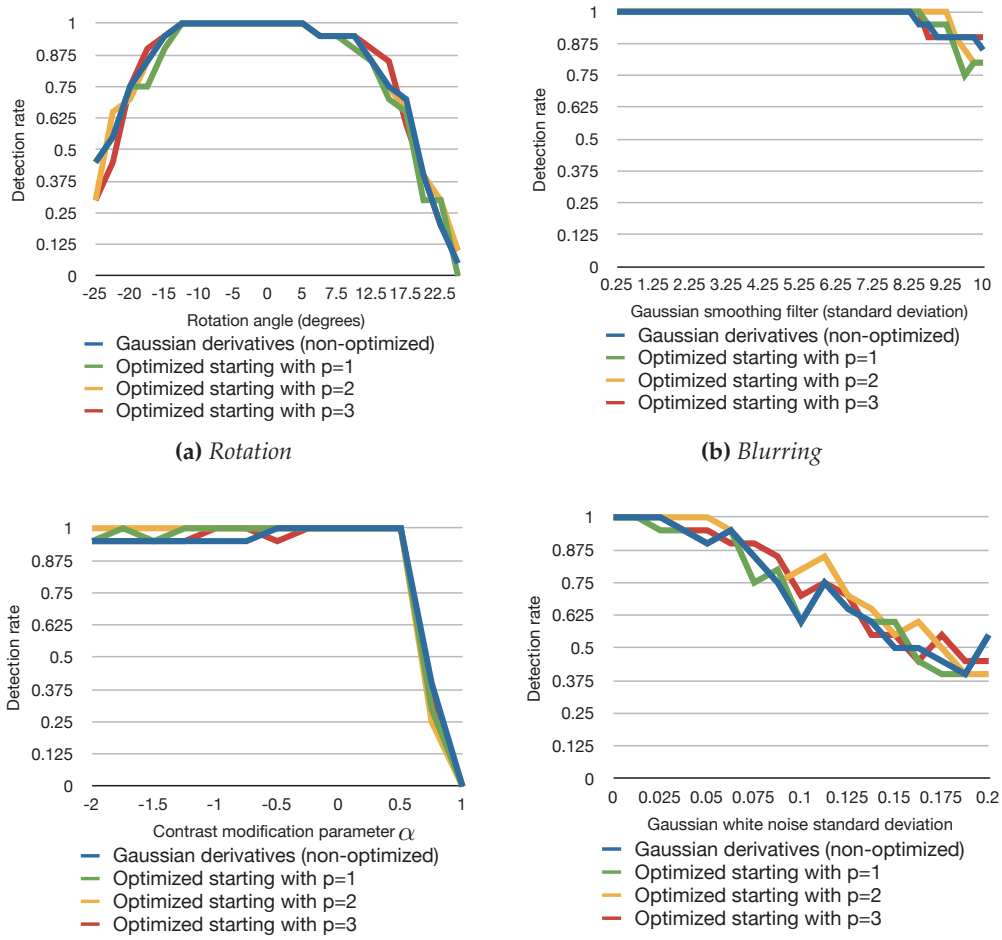
4.6.2 Results on test data sets

In figure 4.14, we present the results of performance using the MIT+CMU face data set, as we can see, the cascades trained using the speed optimized framework continues to operate satisfactorily compared with the non-optimized cascades, in terms of detection rate and false positive rate.

In figure 4.15, results of detection performance for optimized cascades in the FDDB data set are reported using both scores continuous and discrete (see figures 4.15a and 4.15b respectively). In both cases the optimized cascades work with a similar performance than non-optimized cascades.

4.6.3 Results on computational load

In figure 4.16, we report the results of computational load for the speed-optimized cascades. In all cases, despite the similar number of requests made at the pyramid (see figure 4.16b), the optimized cascades increases the number of decision taken (see figure 4.16a), specially in earlier nodes where the number of sub-windows to visit is higher and the number of features is lower. In this experiment is also remarkable how the computational load for the optimized cascades is decreased in almost a half compared with the non-optimized cascades (see figure 4.16c), from the precedent results, we can expect an improvement in detection speed in almost twice compared with the non-optimized cascades. Notice also as the number of nodes necessary to accomplish the learning constraints decreases in one for the optimized cascades trained using $p = 2$. Finally, an example of computational load as function of image is shown in figure 4.17, clearly optimized cascades outperform non-optimized cascades in terms of necessary requests to the Gaussian pyramid.



(a) *Rotation* **(b)** *Blurring*

(c) *Contrast* **(d)** *Noise*

Figure 4.13: *Results of comparing different optimized cascades of Gaussian derivatives in the sensitivity testing dataset. No considerable differences in performance between the non-optimized and the optimized model were noticed.*

4.7 Discussion and Conclusion

In this chapter, we have shown that Gaussian derivatives features can be used to efficiently detect faces in images. Despite the excellent performance of state-of-the-art face detectors in the MIT+CMU face dataset, we shown that Gaussian derivatives outperform in more realistic data sets as the FDDB face data set where the images are similar to these used in our days. In addition, we demonstrate the invariance of cascades of Gaussian derivatives features to image variations as rotation, blurring, noise and contrast using the sensitivity test data set and we have compared all our results with these obtained with a cascade of Haar features which is considered the base line for the face detection approaches.

Furthermore, we have proposed a new speed optimized cascade framework that considered the computational cost of Gaussian derivatives to assign them a correct

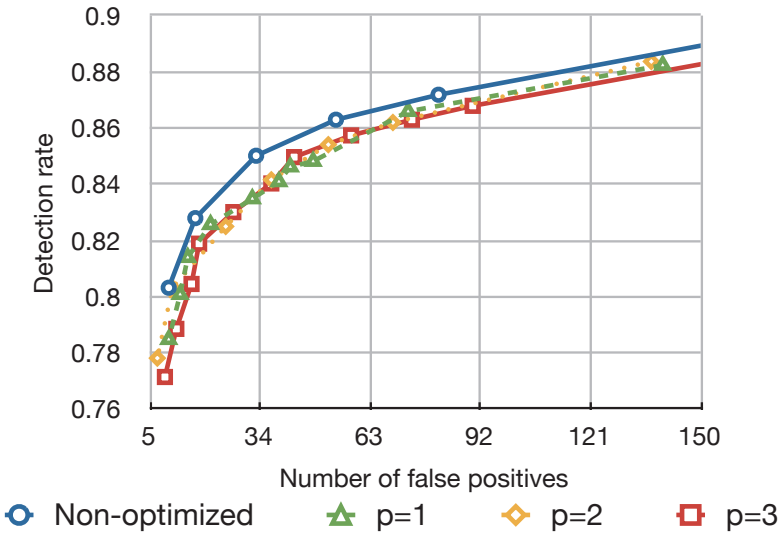
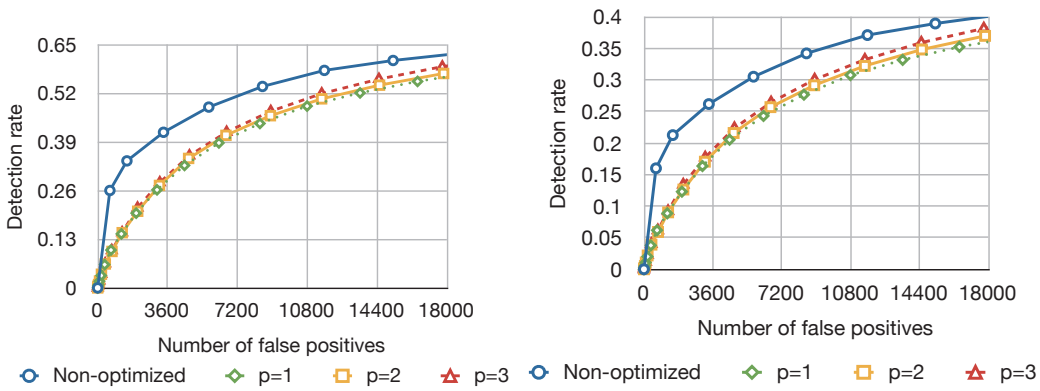


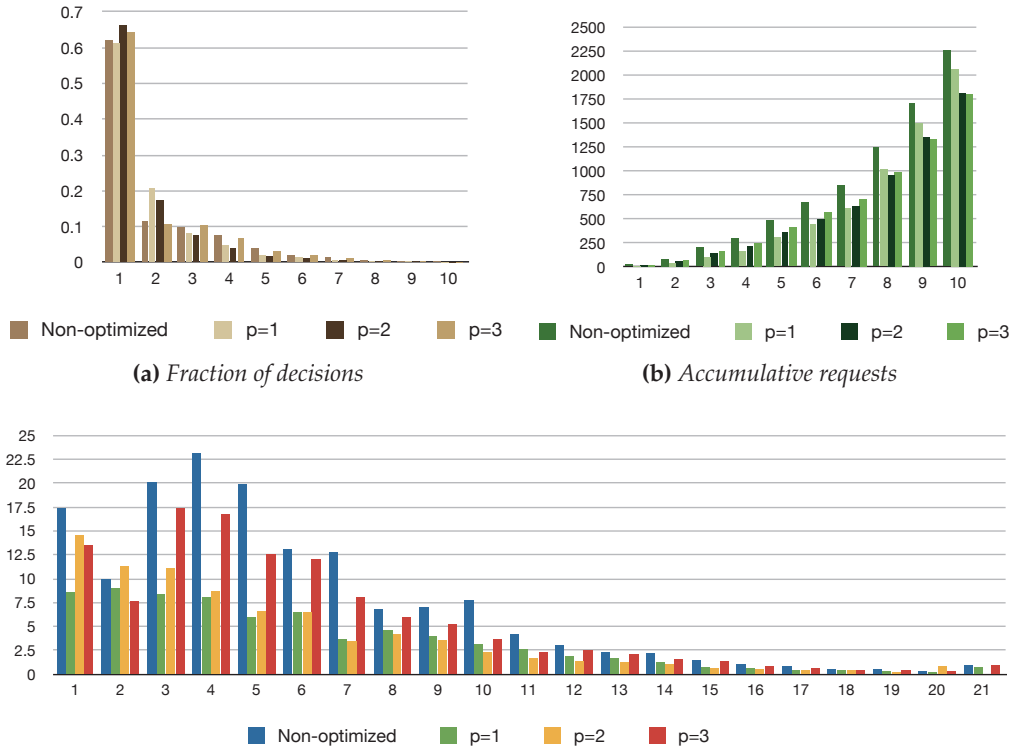
Figure 4.14: *Performance comparison of optimized-speed cascades of Gaussian derivatives features in the CMU+MIT face dataset*



(a) *Discrete Score* **(b)** *Continuous Score*

Figure 4.15: *Performance comparison of speed-optimized cascades of Gaussian derivatives features in the FDDB face dataset*

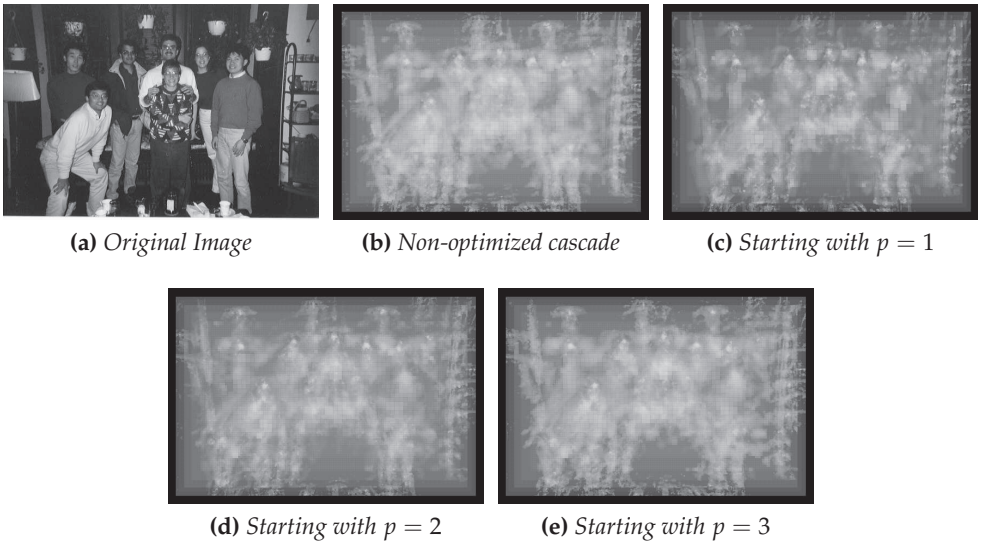
node position in the cascade. (higher order derivatives are positioned in the last nodes where the number of sub-windows is minimum), besides the captured type of information captured for each derivative order is also considered. We have proved with an extensive experimental evaluation that this optimization improves detection speed in almost twice compared with a non-optimized cascade version.



(a) *Fraction of decisions* **(b)** *Accumulative requests*

(c) *Estimated Load*

Figure 4.16: *Computational Load comparisons in the optimized-cascade framework for different values of p*



(a) *Original Image* **(b)** *Non-optimized cascade* **(c)** *Starting with* $p = 1$

(d) *Starting with* $p = 2$ **(e)** *Starting with* $p = 3$

Figure 4.17: *Computational load shown as function of image location for different values of p in the speed-cascade framework*

(a)

(b)

Figure 4.18: *Example of detections using a non-optimized cascade of Gaussian Derivatives. (a) MIT+CMU and (b) FDDB face dataset*

Tenseurs d'Histogrammes de representations Gaussiennes

Dans ce chapitre nous proposons un modelé tensoriel basé en Histogrammes des cartes binaires Gaussiennes pour l'analyse du visage. Le chapitre définit et analyse les différentes étapes pour construire une représentation tensorielle en tenant compte les différentes dimensions possibles comme l'orientation, position et l'échelle. Dans ce chapitre, nous considérons aussi deux différentes architectures tensorielles, la première considère chacun des ordres de dérivés comme un tenseur séparée et le second considère la corrélation entre les dérivés lorsque l'ordre des dérives est ajouté en tant que dimension supplémentaire dans le tenseur finale. Finalement dans le chapitre, l'analysé multilinéaire en composantes Principales est présenté comme un algorithme permettant de réduire les dimensions d'un tenseur sans perte de son structure 3-D en raison de la vectorisation et aussi comme une méthode statistique pour capturer les informations les plus discriminants de chaque dimension considérée dans les tenseurs.

5

Histogram-Tensorial Gaussian Representations

Chapter Contents

5.1 Motivation **73**
5.2 Theoretical Background **74**
5.2.1 What is a Tensor? 74
5.2.2 Multilinear Principal Component Analysis (MPCA) 75
5.3 Histograms of Binary Gaussian Feature Maps **75**
5.4 Tensorial Representations **77**
5.5 Fusing Tensors with MPCA **79**
5.5.1 Individually Tensors 79
5.5.2 Merged Tensors 80
5.6 Summary **80**

5.1 Motivation

As seen in chapter 2(figure 2.1), after a face has been detected, an image representation must be computed for extracting information from the face, to do this, many of the most successful approaches in face recognition (Zhang *et al.*, 2005, 2007; Tan and Triggs, 2007) and aging estimation (Guo *et al.*, 2009) use a space of features based in Gabor Wavelets. These features are combined using statistical tools such as Principal Component Analysis (PCA) (Turk and Pentland, 1991a), Orthogonal Laplacian faces (Fu and Huang, 2008), Discriminative Common Vectors (Cevikalp *et al.*, 2005, 2006), Kernel Locality Preserving Projections with Side Information (KLPPSI) (An *et al.*, 2008), MLASSO (Pham and Venkatesh, 2008) and Volterrafaces (Kumanr *et al.*, 2009).

We find two main disadvantages in the preceding approaches. The first is that they use computationally expensive features such as Gabor wavelets. The second problem is that dimension reduction techniques operate over a feature space of one or two-dimensions. In the case of higher order space feature, this space must be reshaped into vectors. Vectorization breaks the natural structure and correlation in the original feature space (Lu *et al.*, 2008).

To deal with these problems, we have explored the use of a simple set of Gaussian Jet maps calculated with a linear complexity half-octave Gaussian pyramid. Finally, we use a tensorial representation to conserve the spatial structure of the computed feature space and we apply Multilinear Principal Component Analisys (MPCA) (Lu *et al.*, 2008) to reduce the dimensionality in a tensorial fashion.

The rest of the chapter is developed as follows, we present an overview of Multilinear Principal Component analysis in Section 5.2. In Section 5.3, we explain Histograms of Gaussian Binary Maps while section 5.4 presents our tensorial representation. Section 5.5 provides two different methods to apply MPCA in our tensorial representation and some conclusions are presented in Section 5.6.

5.2 Theoretical Background

Tensorial algebra is a huge field in mathematics and physics with a well-developed theory which is out of the scope of the thesis. Nevertheless in the next sections, we are going to present the necessary theoretical background for constructing tensorial representations using Gaussian derivatives.

5.2.1 What is a Tensor?

A Tensor is a multidimensional array, More formally, an N-way or N_{th}-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors. Some examples of real world applications that use tensorial representations include:

- Image Sequences (2D + time)
  - Video
  - Ultrasound
  - Satelite
- Volumes (3D)
  - Magnetic Resonance (MR)
  - Computer Tomography (CT)
- Volume Sequences (3D + time)

– Magnetic Resonance (MR)

In computer vision, tensorial representations have been used with success in gait recognition (Lu *et al.*, 2008; Tao *et al.*, 2007), face recognition (Geng *et al.*, 2011; Yang *et al.*, 2004; Yan *et al.*, 2007b) and visual contents analysis. In this thesis we are going to use tensorial representations only as way to represent feature information without loss of 3-D structure. In computer vision, tensorial representations have been used to represent a sequence of images in its natural space rather than use them for keeping the information structure in a multidimensional feature space.

5.2.2 Multilinear Principal Component Analysis (MPCA)

Multilinear Principal Component Analysis was proposed by Lu *et al.* (2008) as an algorithm of dimensional reduction for tensorial objects. In the category of multilinear subspace learning, MPCA is considered as a tensor-to-tensor projection where the initial tensor is projected to another tensor which contains the most discriminative information captured in the initial tensor. In this section MPCA will be explained and its algorithm exposed.

Let $\{\mathcal{X}_l \in \mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2 \cdots} \bigotimes \mathbb{R}^{I_N}, l = 1, \ldots, L\}$ a set of L tensors examples of N_{th}-order each one. The main objective of MPCA is to find a set of N projection matrices $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, P_n < I_n, n = 1, \ldots, N\}$, such that a projected tensor $\{\mathcal{Y}_l \in \mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2 \cdots} \bigotimes \mathbb{R}^{P_N}, l = 1, \ldots, L\}$ with $\tilde{\mathbf{U}}^{(n)}$ captures most of the variations observed from the original set of tensor samples[1] To solve this, Lu *et al.* (2008) propose the algorithm 4

Once the projected tensors $\mathcal{Y}_m$ have been calculated, each one is rearranged in a vector $\mathbf{y}_l$, ordered in descending order taking into account the computed covariance values in the projected set and only d components by vector are retained with $d \leq L$.

In all our experiments we used a Matlab® implementation of the MPCA algorithm provided by Lu *et al.* (2008)[2]

5.3 Histograms of Binary Gaussian Feature Maps

For the specific task of image classification, a robust representation of image information is desirable. This representation must be invariant to illumination variations and should not be excessively expensive in computational cost. *Histograms of Binary Gaussian Feature Maps (HGBM)* provide such a robust representation and can be used to provide a visual alphabet (Lillholm and Griffin, 2008).

The overall framework to compute HBGM is illustrated in figures 5.2 and 5.1, following the next sequence of operations:

[1] Vectors are denoted by lowercase boldface letters, matrices by uppercase boldface, tensors by calligraphic letters, $\bigotimes$ denotes the Kronecker product and the n-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{U}$ is denoted by $\mathcal{A} \times_n \mathbf{U}$.

[2] http://www.dsp.utoronto.ca/ haiping/index.php

Algorithm 4 MPCA algorithm proposed by Lu *et al.* (2008)

{Giving a set of tensor samples $\{\mathcal{X}_l \in \mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2 \cdots} \bigotimes \mathbb{R}^{I_N}, l = 1, \ldots, L\}$}
{The output is a set of low-dimensional representations $\{\mathcal{Y}_l \in \mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2 \cdots} \bigotimes \mathbb{R}^{P_N}, l = 1, \ldots, L\}$}
Step 1 (Preprocessing): Center the input samples as $\{\tilde{\mathcal{X}}_l = \mathcal{X}_l - \tilde{\mathcal{X}}, l = 1, \ldots, L\}$ where $\tilde{\mathcal{X}} = \frac{1}{L}\sum_{l=1}^{L} \mathcal{X}_l$ is the sample mean.
Step 2 (Initialization): Calculate the eigen-decomposition of $\mathbf{\Phi}^{n_*} = \sum_{l=1}^{L} \tilde{\mathbf{X}}_{l(n)} \cdot \tilde{\mathbf{X}}_{l(n)}^T$ and set $\tilde{\mathbf{U}}^{(n)}$ to consist of the eigenvectors corresponding to the most significant P_n eigenvalues, for $n = 1, \ldots, N$.
Step 3 (Local Optimization):

- Calculate $\{\tilde{\mathcal{Y}}_l = \tilde{\mathcal{X}}_l \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} \ldots \times_N \tilde{\mathbf{U}}^{(N)^T}, l = 1, \ldots, L\}$
- Calculate $\Psi_{\mathcal{Y}_0} = \sum_{l=1}^{L} \|\tilde{\mathcal{Y}}_l\|_F^2$ (the mean $\tilde{\mathcal{Y}}_l$ is all zero since $\tilde{\mathcal{X}}_l$ is centered)
- **for** $k = 1 : K$ **do**
  for $n = 1 : N$ **do**
  Set the matrix $\tilde{\mathbf{U}}^{(n)}$ to consist of the P_n eigenvectors of the matrix $\mathbf{\Phi}^n = \sum_{l=1}^{L} (\mathbf{X}_{l(n)} - \tilde{\mathbf{X}}_{(n)}) \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T \cdot (\mathbf{X}_{l(n)} - \tilde{\mathbf{X}}_{(n)})^T$ where $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} = \tilde{\mathbf{U}}^{(n+1)} \bigotimes \tilde{\mathbf{U}}^{(n+2)} \bigotimes \ldots \bigotimes \tilde{\mathbf{U}}^{(N)} \bigotimes \tilde{\mathbf{U}}^{(1)} \bigotimes \tilde{\mathbf{U}}^{(2)} \bigotimes \ldots \bigotimes \tilde{\mathbf{U}}^{(n-1)}$ corresponding to largest P_n eigenvalues
  Calculate $\{\tilde{\mathcal{Y}}_l, l = 1, \ldots, L\}$ and $\Psi_{\mathcal{Y}_k}$
  if $(\Psi_{\mathcal{Y}_k} - \Psi_{\mathcal{Y}_{k-1}}) < \eta$ **then**
  break and go to step 4

Step 4 (Projection): The feature tensor after projection is obtained as $\{\tilde{\mathcal{Y}}_l = \mathcal{X}_l \times_1 \tilde{\mathbf{U}}^{(1)} \times_2 \tilde{\mathbf{U}}^{(2)} \ldots \times_N \tilde{\mathbf{U}}^{(N)}\}$

1. From an input normalized image a Half-Octave Gaussian Pyramid denoted here by $\mathcal{PYR}$ is build at different levels $\{\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_K\}$.

2. Gaussian features Maps are computed using a bank of t Gaussian Filters (see chapter 3) fixed at different orientations $(\theta_1, \theta_2, \cdots, \theta_m)$ and computed using the above pyramid (same values of σ), this operation is denoted here by:

$$\left\{\mathbf{F}_{(1,\theta_{1:m})}\left(\mathcal{PYR}_{(\sigma_{(1:K)})}\right), \mathbf{F}_{(2,\theta_{1:m})}\left(\mathcal{PYR}_{(\sigma_{(1:K)})}\right), \ldots, \mathbf{F}_{(t,\theta_{1:m})}\left(\mathcal{PYR}_{(\sigma_{(1:K)})}\right)\right\} \tag{5.1}$$

3. A Local Binary Pattern (LBP) Ojala *et al.* (1996); Ahonen *et al.* (2006) is applied over each Gaussian Map to assign a label to each pixel of the image by thresholding the 3×3 neighborhood of each pixel with the center pixel value and considering the result as a binary or decimal number. All the steps above mentioned are illustrated in figure 5.1. Gaussian Binary maps will

be denoted in the rest of the thesis as follows:

$$\left\{\mathbf{M}^{\sigma_{1:K}}_{(1,\theta_{1:m})}, \mathbf{M}^{\sigma_{1:K}}_{(2,\theta_{1:m})}, \ldots, \mathbf{M}^{\sigma_{1:K}}_{(t,\theta_{1:m})}\right\} \tag{5.2}$$

4. We divide each BGM into non-overlapping N rectangular sub-regions with specific size and positions $\{P_1, P_2, P_3, \ldots, P_N\}$ following the next considerations:

   - As the dimension of each level in the Gaussian pyramid is not equal, it is necessary to divide each Binary map corresponding to a level in the pyramid in a sub-set of sub-regions as follows:

$$\begin{aligned} &\left\{\mathbf{M}^{\sigma_1}_{(1:t,\theta_{1:m})}, \mathbf{M}^{\sigma_2}_{(1:t,\theta_{1:m})}, \ldots, \mathbf{M}^{\sigma_K}_{(1:t,\theta_{1:m})}\right\} \\ \equiv &\left\{\left\{\mathbf{M}^{1:P_{n_1}}_{(1:t,\theta_{1:m})}\right\}^{\sigma_1}, \left\{\mathbf{M}^{1:P_{n_2}}_{(1:t,\theta_{1:m})}\right\}^{\sigma_2}, \ldots, \left\{\mathbf{M}^{1:P_{n_K}}_{(1:t,\theta_{1:m})}\right\}^{\sigma_K}\right\} \\ &\text{With } \sum_{k=1}^{K} n_k = N \end{aligned} \tag{5.3}$$

   - finally each sub-region is ranged in a single set denoted by:

$$\left\{\mathbf{M}^{P_1}_{(1:t,\theta_{1:m})}, \mathbf{M}^{P_2}_{(1:t,\theta_{1:m})}, \ldots, \mathbf{M}^{P_N}_{(1:t,\theta_{1:m})}\right\} \tag{5.4}$$

   then a set of histograms with an specific number of bins is computed for each sub-region (see figure 5.2) and denoted as follows:

$$\left\{h\left(\mathbf{M}^{P_{1:N}}_{(1,\theta_{1:m})}\right), h\left(\mathbf{M}^{P_{1:N}}_{(2,\theta_{1:m})}\right), \ldots, h\left(\mathbf{M}^{P_{1:N}}_{(t,\theta_{1:m})}\right)\right\} \tag{5.5}$$

   These histograms encode the most relevant textural and spatial information in manner that is robust to illumination changes.

5.4 Tensorial Representations

In previous works, such histograms were calculated and concatenated to form a single vector (Tan and Triggs, 2007; Zhang *et al.*, 2007; Wu and Rehg, 2008). The inconvenience of this approach is the large size of the final vector as well as the loss of information due to concatenation in only one single vector as each histogram has been calculated in a specific position, for a specific orientation resulting in a loss of the natural 3-D feature space organization. To avoid this, we propose to organize the histograms in a Tensor, thereby conserving the 3-D structure of HBGM, an example of such tensor is shown in figure 5.3.

The tensor corresponding to Gaussian feature t can be denoted by $\mathcal{T}_t \in \mathbb{R}^{N \times M \times bins}$ where N corresponds to number of positions in the construction of

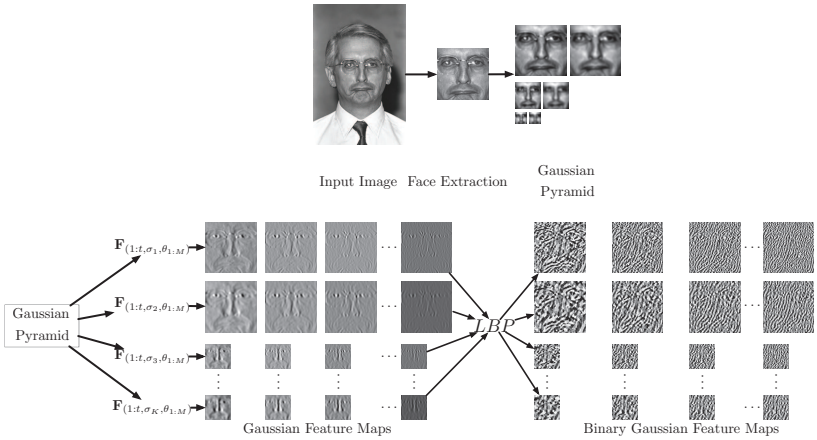


Figure 5.1: *A set of Gaussian filters is computed from the input pyramid then LBP is applied to obtain Binary Gaussian Jet Maps*

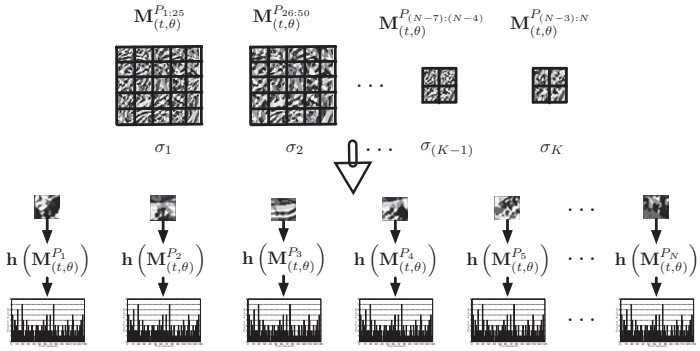


Figure 5.2: *Each local binary map is divided into non-overlapping rectangular sub-regions with a specific size. A set of histograms is then computed for each sub-region*

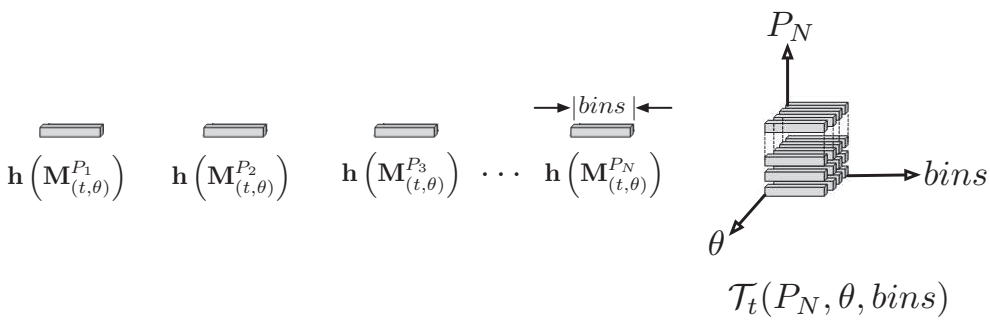


Figure 5.3: *Example of construction for one tensor in our method. Each histogram is arranged in a third order tensor*

histograms, M to the number of orientations used to compute the t feature and *bins*, the number of bins in the computed histogram.

Using this third-order tensor the original feature space is preserved and could be used as an image representation suitable for object recognition and facial analysis, however, the amount of information is still high to be considered in an eventual process of recognition. Following the above idea, it is desirable to find an algorithm for reducing the amount of information available in our feature space, in the next section this idea will be developed using Multilinear Principal Component Analysis.

5.5 Fusing Tensors with MPCA

Recognition in a high-dimensional tensor representation can suffer from:

- "curse of dimensionality", in effect the number of elements in a tensor can be its use intractable in real time applications.
- Wu and Rehg (2008) experimentally demonstrated the correlation in neighboring pixels for Census Transform histograms. This correlation can be interpreted as redundant information. In a similar manner, our tensorial representation computed using LBP is redundant, this may be seen in the correlation between, orientations, bins and positions in our tensorial representations.

To reduce this correlation while conserving the feature spatial distribution of tensors, we propose the use of Multilinear Principal Component Analysis (MPCA) as suggested by Lu *et al.* (2008). Their method determines a multilinear projection that captures most of the original tensorial input variation and supplants existing heterogeneous solutions such as the classical PCA and its 2-D variant 2-D PCA.

Let L the number of tensor samples of each type of feature t and d the number of retained components after applying MPCA, then following this scope, we can apply MPCA in two different ways:

5.5.1 Individually Tensors

MPCA is applied over each tensor $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_t\}$ separetly as shown in figure 5.4a to obtain the vectors $\{y_1, y_2, \ldots, y_t\}$. Finally the resultant vectors are concatenated to form a single vector $y_F \in \mathbb{R}^{t \cdot d}$, where $d \leq L$.

Using this architecture, The features are considered independently without taking into account their class, in particular, Gaussian derivatives of different orders are considered as separate classes without taking into account any possible correlation in the captured information.

5.5.2 Merged Tensors

The tensors $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_t\}$ are concatenated to form an unique tensor of fourth order as shown in the next equation:

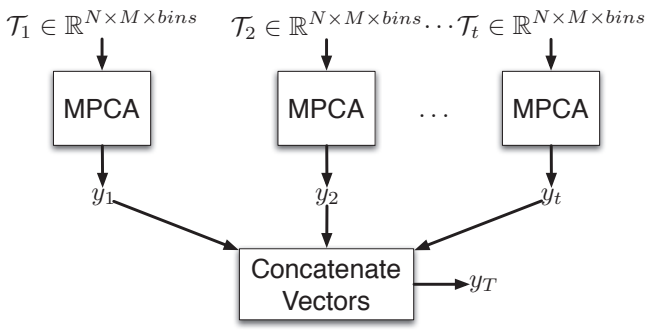
$$\begin{aligned} \mathcal{T}_T &= [\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_t] \\ \mathcal{T}_T &\in \mathbb{R}^{P \times m \times bins \times t} \end{aligned} \tag{5.6}$$

MPCA is then applied to this tensor $\mathcal{T}_T$ to obtain a vector $y_T \in \mathbb{R}^d$, with $d \leq L$. The above process is illustrated in figure 5.4b.

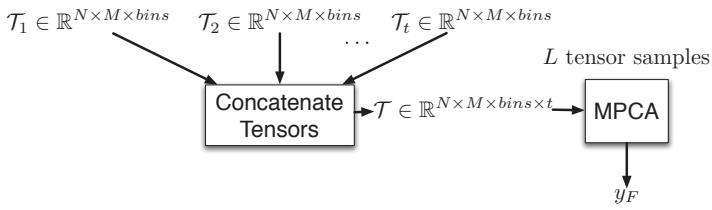
In this architecture, the correlation between feature-classes is considered and the resultant vector is smaller, however the tensor-order increases in one order making the MPCA process more complex

5.6 Summary

We have introduced a new image representation model that uses a simple set of Gaussian filters calculated effectively with a Half-Octave Gaussian pyramid and a tensorial representation that conserves the natural structure of the feature space described for such filters. Two algorithmic structures for fusing tensors using MPCA have been proposed. Each one of these structures will be applied in sub sequent chapters to solve the problem of face recognition and age estimation showing the performance of these representations to describe human facial appearance.



(a) *Concatenating vectors*



(b) *Concatenating Tensors*

Figure 5.4: *Fusing tensors with MPCA. (a) MPCA is applied at each tensor and then the resulting vectors are fused into one. (b) The tensors are fused into one before applying MPCA*

Reconnaissance de visages avec les tenseurs des HBGM

Décrit comme des représentations tensorielles basées sur des dérivés gaussienne (tenseurs de HBGM) sont appliquées au problème de reconnaissance de visage. Dans ce chapitre, l'analyse multilinéaire en composantes principales est utilisé pour réduire la dimension spatiale du tensor et Kernel Discriminative Common Vectors (KDCV) est utilisé pour améliorer les résultats de reconnaissance. De façon plus détaillée ce chapitre est divisé en trois parties principales, la première fait une présentation brève des KDCV. Dans la deuxième partie les dérivés de premier et second ordre sont utilisés dans notre représentation tensorielle. Cette partie de la thèse a été liée aux résultats de notre première représentation tensorielle et seulement le vecteur final y_T a été utilisé. Dans la dernière partie de ce chapitre trois types de caractéristiques basées dans les dérivés gaussiens sont considérés dans notre représentation tensorielle: *Mag* (magnitude du gradiente), *log* (Laplacien de Gaussiennes) et γ (la troisième composante de la norme du Gaussian Jet).

Dans tout le chapitre, trois publics disponibles face à des ensembles de données (Feret, Yale et de Yale B + Extended Yale) sont utilisés pour valider l'approche.

-Expériences dans le visage FERET ensemble de données en utilisant les caractéristiques des dérivés de Gauss montrent premier et second ordre que notre approche est compétitive (performances similaires) avec l'état d'autres approches de l'art de l'usage que les ensembles de fonctionnalités plus complexes (voir la section 6.4.1). En plus des résultats dans l'ensemble de données de Yale en utilisant des dérivés gaussiens fonctions jusqu'à la deuxième commande de façon tensorielle surpasser d'autres approches dans près de 1 % (voir la section 6.4.2). Enfin les résultats dans le Yale Yale B + étendues peuvent être observées dans la section 6.4.3, ce résultat révèle une invariance élevé de notre méthode à des variations d'illumination.

-Résultatsdans le jeu de données en utilisant FERET *Mag* (magnitude de gradient), *log* (Laplacien de Gaussiennes) et γ (le troisième volet de la seconde locales afin de Gauss norme jet) ainsi que son coût de calcul sont présentés dans les sections 6.5.4 et 6.5.1 respectivement, une fois encore les dérivés gaussiens obtenir des résultats comparatifs ou supérieure à d'autres approches dans le state-of-the-art. En outre nous comparons également les performances de chaque vecteur résultant en utilisant les deux configurations tensoriel proposé dans cette thèse (voir les sections 6.5.2 et 6.5.3). Enfin les résultats dans le Yale B + données étendue de Yale sont présentés dans la section 6.5.5.

6

Face Recognition using Tensors of HBGM

Chapter Contents

6.1 Motivation **85**
6.2 Theoretical Background **86**
6.2.1 Kernel Discriminative Common Vectors (KDCV) 86
6.2.2 Experimental datasets 86
6.3 Face Recognition Architectures **89**
6.3.1 Preprocessing of face images for recognition 90
6.4 Results using First and Second Order Derivatives **90**
6.4.1 Results in the FERET face dataset 91
6.4.2 Results in the YALE dataset 92
6.4.3 Results in the Yale B + Extended Yale B Face Dataset 92
6.5 Results using Mag, LoG **and** γ **92**
6.5.1 Computational Cost 93
6.5.2 Performance of y_1, y_2 and y_3 93
6.5.3 Discriminant Capacity of y_T and y_F 94
6.5.4 Results in the FERET face dataset 95
6.5.5 Results in the Yale B + Extended Yale B Face Dataset 96
6.6 Summary and Conclusion **98**

6.1 Motivation

THIS chapter describes the design and implementation choices made in the development of our face recognition system using Histograms of Binary Gaussian maps (HGBM) in a tensorial representation. As mentioned in chapter 5 our tensorial representation conserves the 3D feature space structure and considers

multiple types of features, these properties before mentioned are desirable in the complicated process of facial analysis, specially this of face recognition.

To explain our proposed solution for recognizing faces, we organized as follows, In section 6.2, we present a review of Kernel Discriminative Common vectors as a method to improve recognition performance. In sections 6.2.2 through 6.3, we describe two different architectures for recognizing faces and we present our experimental set up for testing HBGM in these proposed architectures. Experimental results using different types of features computed with a half-octave pyramid are presented in sections 6.4 and 6.5. Finally we present some conclusions in section 6.6.

6.2 Theoretical Background

6.2.1 Kernel Discriminative Common Vectors (KDCV)

In face recognition, the discriminative power of each final vector y_T or y_F can be improved by projection onto an optimal discriminative space with a KDCV Cevikalp *et al.* (2006) (Kernel Discriminative Common Vector). This kernel method has been proved with success in Tan and Triggs (2007). The characteristic equation for the KDCV is:

$$l_{test} = \left(U\Lambda^{1/2}V\right)^T k_{test} \quad l_{test} \in \Re^t \tag{6.1}$$

Where Λ is the diagonal matrix with non-zero eigenvalues, **U**, the associated matrix of normalized eigenvectors, **V** is the basis for the null space of the projected within-class scatter matrix and k_{test} is a vector with entries $K(y_i^t, y_{test}) = \langle \phi(y_i^t), \phi(y_{test}) \rangle$, where $\phi(y_i^t)$ are the mapped training samples in a high dimensional space and $K()$ is a typical kernel, for details see Cevikalp *et al.* (2006); Tan and Triggs (2007).

The Discriminative Common Vectors with Kernels (KDCV) method can be summarized in the algorithm 5

6.2.2 Experimental datasets

Three public available face data sets were used to test our face recognition methods, these data sets are the FERET , Yale and Yale B+ Extended Yale B.

6.2.2.1 FERET Dataset

The FERET database (Phillips *et al.*, 2000) was collected in 15 sessions between August 1993 and July 1996, and it contains a total of 14,126 images from 1,199 individuals with views ranging from frontal to left and right profiles. The face

Algorithm 5 Kernel Discriminative Common Vectors (Cevikalp *et al.*, 2006).

{Giving a Matrix $\Phi = [\phi(x_1^1)\phi(x_1^2)\ldots\phi(x_{N_1}^1)\phi(x_1^2)\ldots\phi(x_{N_C}^C)]$ whose columns are the transformed training samples in a higher dimensional space $\Im$}.
{ Let the within-class scatter matrix S_W^Φ, the between class scatter matrix S_B^Φ and the total scatter matrix S_T^Φ computed from the Φ training set}

1. project the training set samples Φ onto a more discriminative space $R(S_T^\Phi)$ (defined as the null space of the total scatter matrix) through the Kernel PCA. Let

   $\tilde{K} = K - 1_M K - K 1_M + 1_M K 1_M = U \Lambda U_T$

   Where Λ is the diagonal matrix of nonzero eigenvalues, U is the matrix of normalized eigenvectors associated to Λ , M the number of total samples and $1_M \in \mathcal{R}^{M\times M}$ is a matrix with entries $\frac{1}{M}$. Here the kernel matrix $K \in \mathcal{R}^{M\times M}$ is given by $K = \Phi^T\Phi = \left(K^{ij}\right)_{\substack{i=1,\ldots,C\\ j=1,\ldots,C}}$, where each matrix $K^{ij} \in \mathcal{R}^{N_i\times N_j}$ is defined as

   $K^{ij} = (k_{mn}^{ij})_{\substack{i=1,\ldots,C\\ j=1,\ldots,C}} = \langle\phi(x_m^i), \phi(x_n^j)\rangle = k(x_m^i, x_n^j)_{\substack{m=1,\ldots,N_i\\ n=1,\ldots,N_j}}$

   where $k(\cdot)$ represents the kernel function

2. Compute the new total within-scatter matrix on the new reduced space as follows:

   $\tilde{S}_W^\Phi = \Lambda^{\frac{1}{2}} U^T \tilde{K}_W \tilde{K}_W^T U \Lambda^{\frac{1}{2}}$

   where $\tilde{K}_W = (K - 1_M K)(I - G)$ and $G = diag[G_1, \ldots, G_C] \in \mathcal{R}^{M\times M}$ is a block-diagonal matrix and each $G_i \in \mathcal{R}^{N_i\times N_i}$ is as matrix with all its elements equal to $\frac{1}{N_i}$

3. Find vectors that span the null space of $\tilde{S}_W^\Phi$ by eigen-decomposition solving the equation $V^T \tilde{S}_W^\Phi V = 0$

4. {The output are the matrices V, Λ and U}

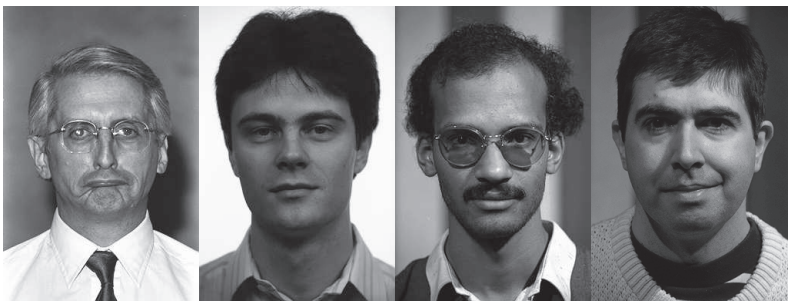


Figure 6.1: *Examples of images from the FERET data set*

Figure 6.2: *Example of images from the Yale face database*

images were collected under relatively unconstrained conditions. The same physical setup and location was used in each session to maintain a degree of consistency throughout the database. However, since the equipment was reassembled for each session, images collected on different dates have some minor variation. Sometimes, a second set of images of an individual was captured on a later date, resulting in variations in scale, pose, expression, and illumination of the face. Furthermore, for some people, over two years elapsed between their first and last capturing in order to study changes in a subject's facial appearance over a year. An example of images from the FERET data set are shown in figure 6.1

The testing protocol in the FERET data set is performed using four probe sets: the 'fb' set contains faces with variation in expression, the 'fc' set with lighting variation and the 'dup I' and 'dupII' sets contain variation dues to aging of the subject. We used the FERET distributed training set plus the gallery images 'fa' to train the KDCV. As shown in Tan and Triggs (2007), the addition of the gallery increases the dimensionality of the final discriminative space. We have compared our results with the best results in the FERET'97 test Phillips *et al.* (2000) and the published results in Zhang *et al.* (2007, 2005); Tan and Triggs (2007); Lui and Beveridge (2008). The rank-1 recognition rates of different methods on the FERET probe sets are show in Table 1. To our knowledge, the results in Tan and Triggs (2007) and Lui and Beveridge (2008) are the most recent state-of-the-art with the FERET database.

6.2.2.2 Yale Face Dataset

The Yale face data set [1] was constructed at the Yale Center for Computational Vision and Control to evaluate robustness to variations in facial expression.

The YALE face database contains 165 images of 15 individuals under various facial expressions (with glasses, happy, without glasses, normal, sad, sleepy, surprised and wink) and lighting conditions (center, right and left light). Some example images are shown in figure 6.2.

6.2.2.3 Yale B + Extended Yale B

The Yale B + Extended Yale B (Georghiades *et al.*, 2001) [2] is composed of two different testing data sets. The Yale face dataset B contains images obtained

[1] The Yale Face Dataset, `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`, Accessed: April 2009

[2] `http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`

from 10 individuals. Images are captured under 64 different lighting conditions from 9 pose views and are divided into 5 subsets according to the ranges of the illumination angles between the light source direction and the camera axis. An example of images from this data set is shown in figure 6.3. The extended Yale-B dataset contains 16128 images of 28 human subjects captured under the same condition as Yale B. In our experiments, we use only the frontal face images from these two databases.

6.3 Face Recognition Architectures

Two different methods using tensors of HGBM are proposed, the first one use the vector y_F and the last one uses y_T. In both cases, one of these vectors is considered as input to a trained KDCV kernel method and its output is compared with a gallery set using a simple Nearest Neighborhood algorithm. The general diagram for the methods before mentioned is presented in figure 6.4

To calculate histograms, we used a sub-region size of 16×16 pixels (a border of 8 pixels in each image is left untested for faces to avoid problems related with image borders), removing two bins corresponding to values of 0 and 255 per each histogram. The remaining bins are grouped to form a 127 bin histogram. To compute the orthogonal matrix in the MPCA method for all our experiments, we used the FERET distributed training set and we retained only 1000 entries (l=1000) per tensor. After applying MPCA, each final vector y_1, y_2 and y_3 as well as y_T and y_F is normalized to an unit standard deviation.

The final two architectures for recognizing faces are shown in figure 6.4 The discriminative power of each vector y_t or y_f can be improved using KDCV, in this case, we used a Gaussian kernel $k(x, y) = e^{-\|x-y\|^2/q}$ with a scale parameter q chosen experimentally to obtain the best results (the value of q is reported in each experiment). Finally the discriminative vector calculated by KDCV is classified into one of the vectors in the gallery set using the nearest neighbor rule and a cosine distance.

Figure 6.3: *Example of images from the Yale B + Extended Yale B dataset*

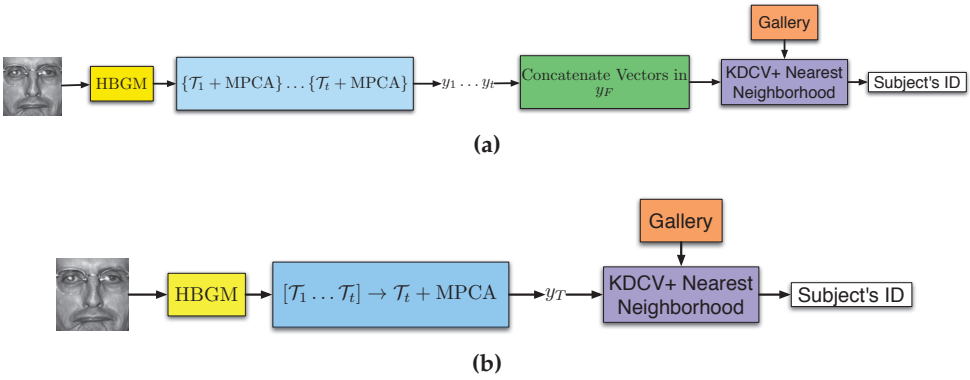


Figure 6.4: *Face recognition architectures proposed in this thesis using tensors of HBGM. (a)concatenated vectors then KDCV+NN. (b) concatenated tensors then MPCA and finally KDCV+NN*

6.3.1 Preprocessing of face images for recognition

In this research, only gray-level facial images are considered without taking color information into account. First, all color images are transformed to gray-level images by taking the luminance component in the Y CbCr color space. Then, all face images are rotated and scaled so that the centers of the eyes are placed on specific pixel position using the manually annotated coordinate information of eyes. Next, the image is cropped and normalized to 128×128 pixels, followed by histogram equalization, and image intensity values are normalized to have zero mean and unit standard deviation. Finally, each image is represented with 256 gray levels (eight bits) per pixel.

6.4 Results using First and Second Order Derivatives

In this experiments Gaussian Binary maps (see chapter 5) are obtained by computing Gaussian derivative features of first and second order at four different orientations $\theta = \{0, \frac{\pi}{4}, \frac{\pi}{2}, 3\frac{\pi}{2}\}$ at 6 levels from the half-octave pyramid, corresponding to $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 8\}$. We used also the feature maps from the 0th order Gaussian pyramid. This descriptor provide a textural description that complements the information given by Gaussian Derivatives Features. We denote this feature as follows:

$$\begin{aligned}
\mathbf{F}_{(1,\theta=\{0,\frac{\pi}{4},\frac{\pi}{2},3\frac{\pi}{2}\})} &= G_{1,\theta}\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 8\}\right) \\
\mathbf{F}_{(2,\theta=\{0,\frac{\pi}{4},\frac{\pi}{2},3\frac{\pi}{2}\})} &= G_{2,\theta}\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 8\}\right) \\
\mathbf{F}_{(3,\theta=\{0\})} &= G(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 8\})
\end{aligned} \tag{6.2}$$

The total number of Gaussian maps calculated in this experiment is 66. For Gaussian derivatives features of first order we compute 24 Gaussian maps (4 orientations and 6 levels), for Gaussian derivatives features of second order we compute 30 Gaussian maps (4 orientations, 6 levels plus the maps of G_{xy}) and

finally we use 6 Gaussian maps resultant from original scale space in the first six levels of the pyramid.

To build the HGBM, we separate the Gaussian derivatives above mentioned in three different tensors defined as follows:

- $\mathcal{T}_1 \in \mathbb{R}^{118\times4\times127} \rightarrow$ Histograms of Binary Gaussian derivative features of first order,
- $\mathcal{T}_2 \in \mathbb{R}^{118\times5\times127} \rightarrow$ Histograms of Binary Gaussian derivatives features of second order plus the maps for G_{xy},
- $\mathcal{T}_3 \in \mathbb{R}^{118\times127} \rightarrow$ Histograms of Binary maps from the original Gaussian scale space.

From the above tensors, we conducted experiments in the FERET, Yale and Yale B + Extended Yale data sets using only the output vector y_T.

6.4.1 Results in the FERET face dataset

The rank-1 recognition rates of different methods on the FERET probe sets are show in Table 6.1 . To our knowledge, the results in Tan and Triggs (2007) and Lui and Beveridge (2008) are the most recent state-of-the-art with the FERET database. Our results with the FERET database are statistically equivalent (with a difference of ±0.01) or better to the most recent state-of-the-art results on this dataset. Note that most of the methods described in Zhang *et al.* (2007) Zhang *et al.* (2005) and Lui and Beveridge (2008) use the Gabor wavelets to generate their maps. These wavelets have a much higher algorithmic complexity and overall computing cost that is not improvable. On the other hand, our Gaussian derivatives features calculated with the half-octave pyramid to generate feature maps Crowley and Riff (2003), can be provided with a fast linear complexity algorithm, and are thus much more suitable for real applications.

	FERET Probe Sets			
Method	$\mathbf{f_b}$	$\mathbf{f_c}$	$\mathbf{Dup_I}$	$\mathbf{Dup_{II}}$
Best Results (Phillips *et al.*, 2000)	0.96	0.82	0.59	0.52
LGBPHS_Weighted (Zhang *et al.*, 2005)	0.98	0.97	0.74	0.71
HGPP_Weighted (Zhang *et al.*, 2007)	0.97	0.99	0.80	0.78
Gabor+LBP_KDCVM (Tan and Triggs, 2007)	0.98	0.98	0.90	0.85
GRM-Local (Lui and Beveridge, 2008)	0.98	0.98	0.80	0.84
$y_T(G_1, G_2, G)$ + KDCV	**0.98**	**0.98**	**0.90**	**0.85**

Table 6.1: *The Rank-1 Recognition Rates of Different Algorithms on the FERET Probe Sets*

6.4.2 Results in the YALE dataset

In this experiment, Images with a neutral facial expression are used as gallery set. We augment the gallery set with images without glasses to train the Kernel DCV and and We used the same Orthogonal matrix used in the precedent dataset. The remaining images are used as a probe set to compare our results with the results presented in (Cevikalp *et al.*, 2005; Yang *et al.*, 2004). The rank-1 recognition rates of different methods on the YALE dataset are shown in the Table 6.2 Clearly the results of the proposed method are better than the best results reported in Cevikalp *et al.* (2005) and Yang *et al.* (2004), demonstrating reliability of our method under changes in facial expression.

6.4.3 Results in the Yale B + Extended Yale B Face Dataset

The images with the most neutral illumination(denoted as A+000E+00 in this data set) are used as gallery set. To train the Kernel DCV we take the images in the gallery set and a randomly selected image from the first subset. We did not test our method with the challenging subset 5 and We used the same Orthogonal matrix used in the precedent dataset. The rank-1 recognition rates of different methods on the Yale B + extended Yale B dataset are show in the Table 6.3. In this case we outperformed the published results in Xie *et al.* (2008) for the subset number 3 and we have the same results for the subsets 1 and 2. Note that while the results reported in Xie *et al.* (2008) are for different methods of illumination normalization, we do not normalize the images for changes in illumination and our method does not try to specifically solve this problem.

6.5 Results using Mag, LoG and γ

In this experiments Gaussian Binary maps (see chapter 5) are obtained by computing Mag(gradient magnitude), LoG (Laplacian of Gaussians) and γ (the third component of the second local order Gaussian jet norm) at 4 levels from the half-octave pyramid, corresponding to $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}$. We denoted these features as follows:

Method	Recognition Accuracy
Eigenfaces (Cevikalp *et al.*, 2005)	76.0%
2DPCA (Yang *et al.*, 2004)	84.2%
Fisherfaces (Cevikalp *et al.*, 2005)	96.0%
DCVM (Cevikalp *et al.*, 2005)	97.3%
$y_T(G_1, G_2, G)$ + KDCV	**98.2%**

Table 6.2: *The Rank-1 Recognition Rates of Diferent Algorithms on The YALE Face Dataset*

Method	Subset Number 1	2	3	4
LTV (Xie *et al.*, 2008)	100.0%	99.8%	79.4%	76.1%
RLS(LOG-DTC) (Xie *et al.*, 2008)	100.0%	100.0%	87.1%	87.6%
$y_T(G_1, G_2, G)$ + KDCV	**100.0%**	**100.0%**	**94.7%**	**60.1%**

Table 6.3: *The Rank-1 Recognition Rates of Different Algorithms on The YALE B+EXTENDED YALE-B Face Dataset*

$$\begin{aligned}
\mathbf{F}_{(1,\theta=\{0\})} &= Mag_\theta\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}\right) \\
\mathbf{F}_{(2,\theta=\{0\})} &= LoG_\theta\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}\right) \\
\mathbf{F}_{(3,\theta=\{0\})} &= \gamma_\theta(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4, \})
\end{aligned} \tag{6.3}$$

The total number of Gaussian maps calculated in this experiment is 12, an example of such maps with different conditions of illumination is shown in figure 6.5.

To build the HGBM, we separate the Gaussian derivatives above mentioned in three different 2nd order tensors defined as follows:

- $\mathcal{T}_1 \in \mathbb{R}^{116\times127} \rightarrow$ Histograms of Binary Gaussian maps of Mag,
- $\mathcal{T}_2 \in \mathbb{R}^{116\times127} \rightarrow$ Histograms of Binary Gaussian maps of LoG,
- $\mathcal{T}_3 \in \mathbb{R}^{116\times127} \rightarrow$ Histograms of Binary Gaussian maps of γ.

6.5.1 Computational Cost

To evaluate the computational cost, we recorded the average CPU time for each step in the algorithm using a 2.0 GHz dual core PC. Each algorithm was applied to over 2000 images selected randomly from the FERET dataset as shown in table. 6.4. Notice that the average CPU time is less than 2 seconds in a non-optimized MATLAB® implementation.

6.5.2 Performance of y_1, y_2 and y_3

In order to quantify the discriminative power of each Gaussian map in this experiment, we compute the tensorial representation from each Gaussian binary map as shown in section 5.4 and then we apply MPCA to obtain y_1, y_2 and y_3. This procedure was done for each probe set of the FERET dataset, and the results of recognition accuracy versus dimensionality reduction and the cumulative match curves are shown in figures 6.6 and 6.7. This figure indicates that y_1 with a lower dimension outperforms in the sets f_b and dup_{II} but have a poor discriminative

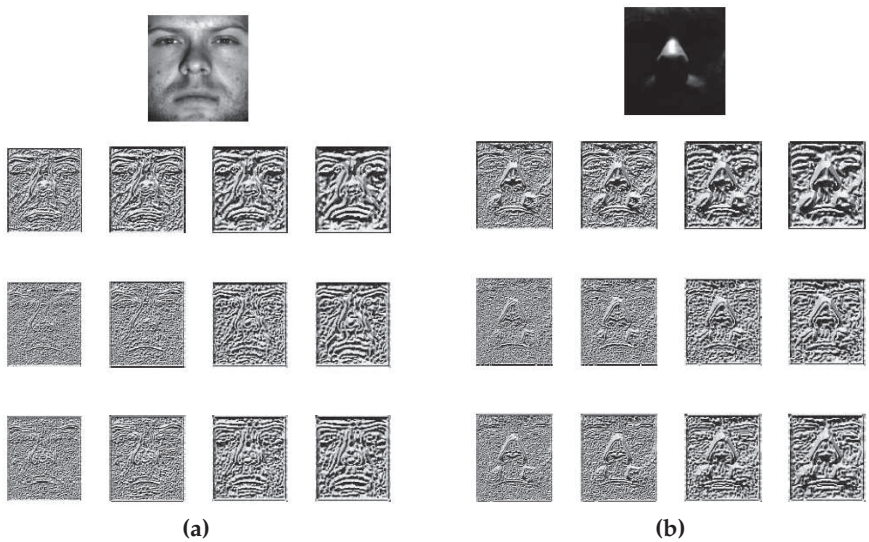


Figure 6.5: *Binary Gaussian maps at different scales (each column) of two images with different conditions of illumination ((a) and (b)), the three last rows correspond to* $Mag_\theta\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}\right)$, $LoG_\theta\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}\right)$ *and* $\gamma_\theta(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4, \})$ *respectively*

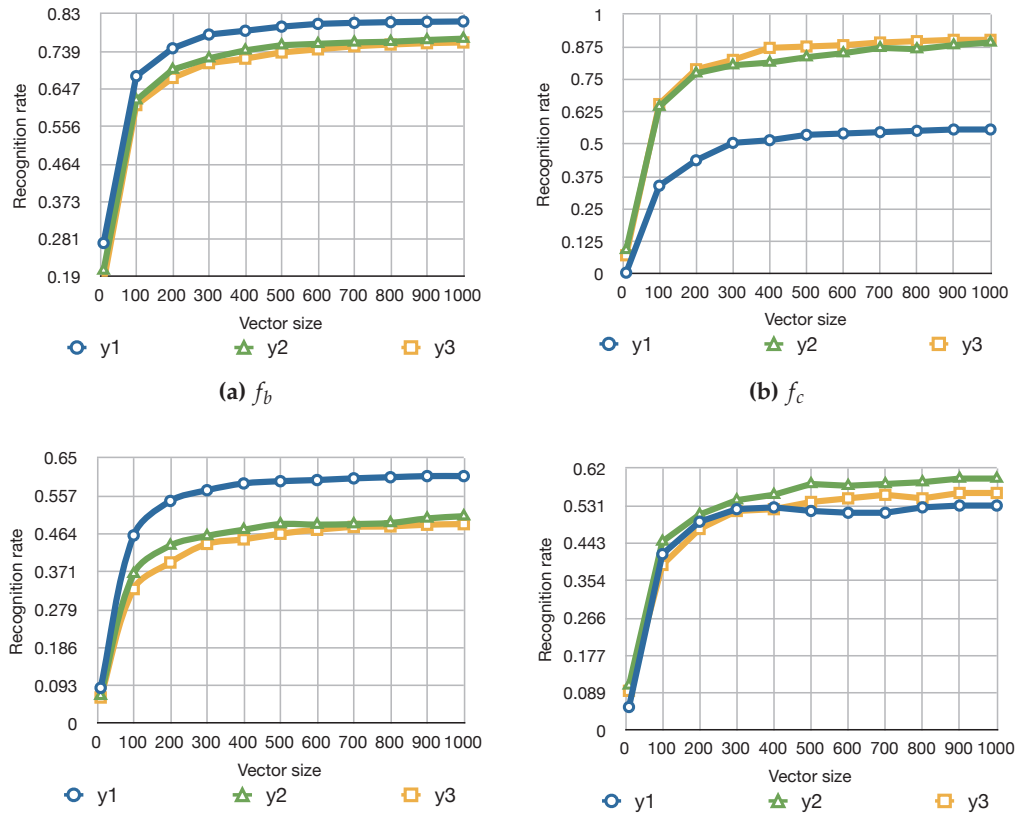
Table 6.4: *CPU average times of each step in our face recognition method using* Mag, LoG *and* γ

Pipeline Step	CPU Time (s)
Half-Octave Pyramid	0.003
Mag	0.30
Log	0.52
γ	0.60
$\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_3$	0.10 (each)
MPCA Projection	0.01
Total Time	**1.86**

power in the set f_c. These results show a certain robustness to age variation but a weakness related to illumination variations. Vectors y_2 and y_3 outperform other vectors with a highly score for the subset f_c. This result shows good robustness with illumination variations and with age variations. Finally for the subset dup_{II} all the vectors perform similarly with a little advantage for y_3. These results also show a highly complementarity for each binary Gaussian map according to the type of variation on the image. This assumption is tested in the next experiment.

6.5.3 Discriminant Capacity of y_T and y_F

We have experimentally evaluated the performance of each face recognition algorithm without KDCV. In this case we sought to compare the results of concatenating the tensors and then MPCA or applying MPCA to each tensor



(a) f_b **(b)** f_c **(c)** Dup_I **(d)** Dup_{II}

Figure 6.6: *Cumulative Match Curves for the vectors y_1, y_2 and y_3 using Mag, LoG and γ on the four FERET probe sets (a)f_b (b) f_c, (c) Dup_I and (d)Dup_{II}*

and then concatenating the vectors y_n. We computed the vectors y_F and y_T as it described in section 5.5, over each probe set in the FERET dataset. The results of recognition accuracy versus dimensionality reduction are shown in figure 6.8. These observations show that in the case of lower dimensions $m \leq 600$ vector Y_T outperforms Y_F in all the sub-sets, but for higher dimension $m \geq 600$ both vectors provide similar performances. In other hand, in figure 6.9 the performance of y_F and y_T show a similar behavior in all the four subsets with a low superior performance for y_T in all the four probe sets.

6.5.4 Results in the FERET face dataset

We have evaluated the proposed method using all four FERET probe sets. To train KDCV we used the distributed training set plus the gallery images "fa", as shown in Tan and Triggs (2007). The addition of the gallery increases the dimensionality of the final discriminative space. We employed a small set of randomly created training and test sets to compute the best Gaussian parameter q. These datasets were only used for parameter selection and were not employed for further tests.

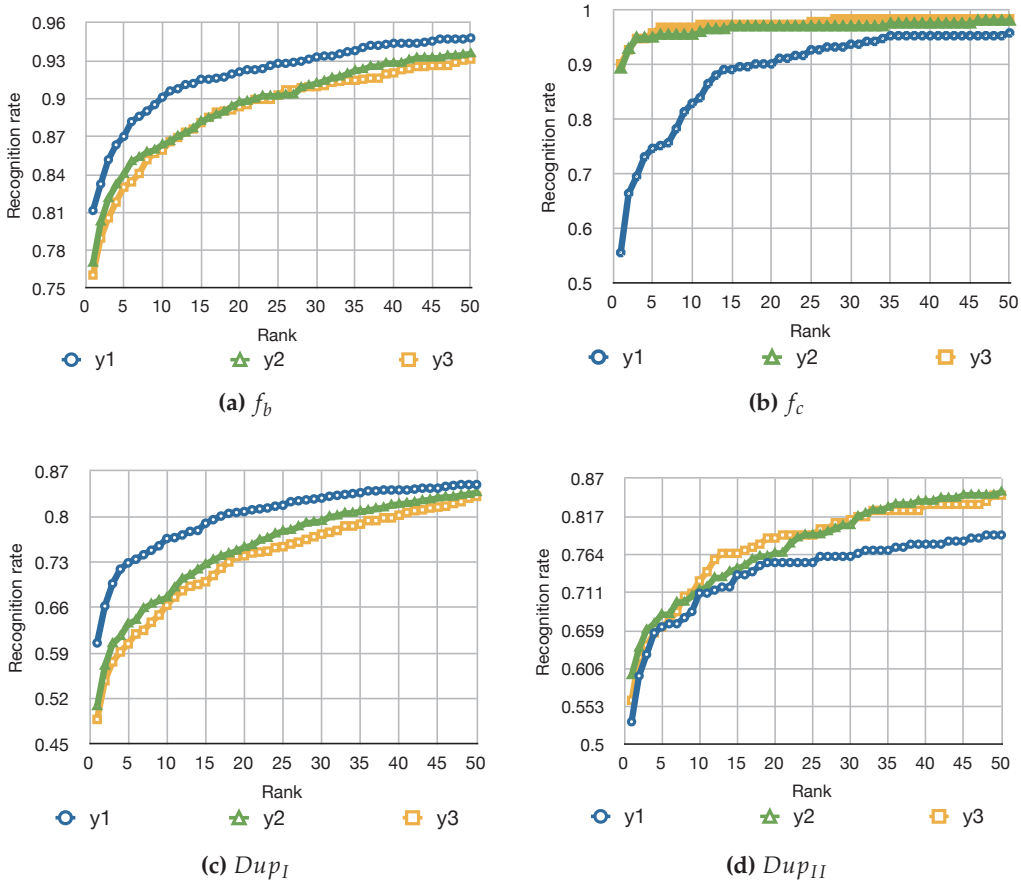
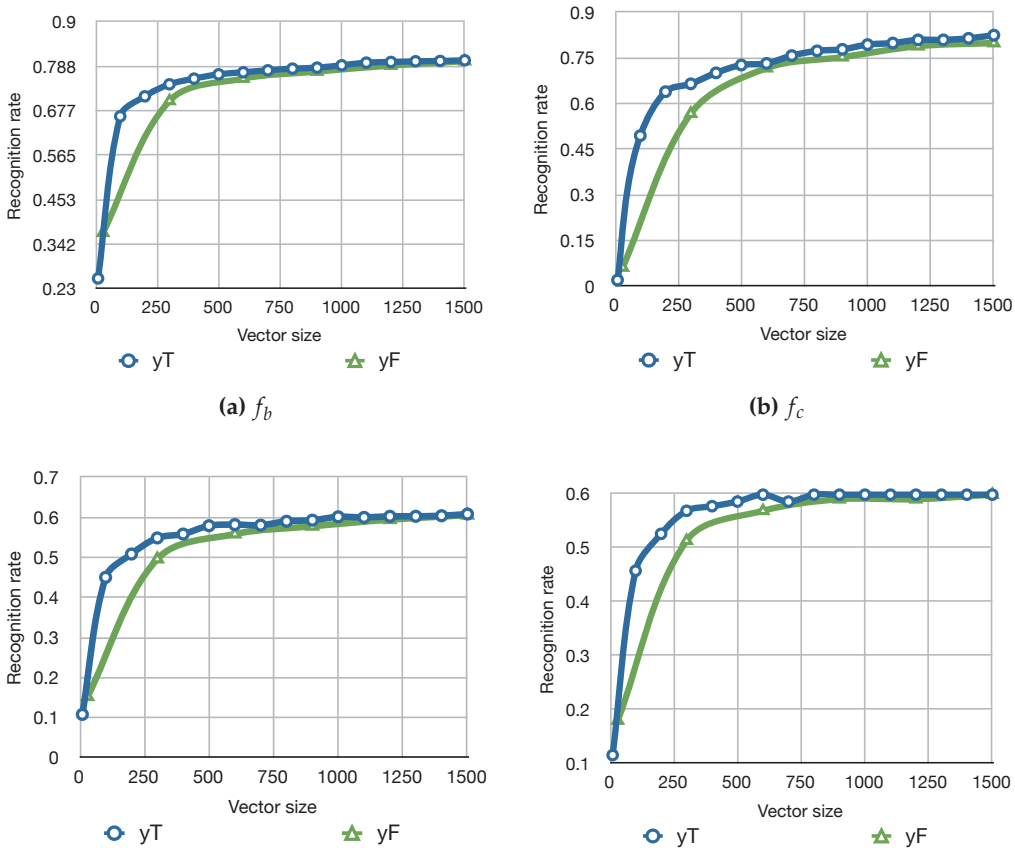


Figure 6.7: *Rank-1 recognition accuracy versus dimensionality reduction with MPCA for the vectors y_1 , y_2 and y_3 using Mag, LoG and γ on the four FERET probe sets (a) fb , (b)fc, (c)dup I and (d)dup II.*

We have compared our results with the best results in the FERET'97 test Phillips *et al.* (2000) and the published results in Zhang *et al.* (2007, 2005); Tan and Triggs (2007); Lui and Beveridge (2008). The rank-1 recognition rates of different methods on the FERET probe sets are show in Table 6.5. To our knowledge, the results in Tan and Triggs (2007) and Lui and Beveridge (2008) are the most recent state-of-the-art results with the FERET database.

6.5.5 Results in the Yale B + Extended Yale B Face Dataset

We test also our face recognition architectures using the yale B + extended Yale datasets to show the invariance to illumination of our method. We used the same Orthogonal matrix used in the precedent dataset and the same Gaussian parameter q. In this dataset, each experiment was repeated 10 times for 10 random choices of the training set for KDCV. All images other than the training set were used for testing, this testing procedure is the same used in Cai *et al.* (2007); Hua *et al.* (2007); An *et al.* (2008); Fu and Huang (2008); Kumanr *et al.* (2009). We have reported



(a) f_b **(b)** f_c **(c)** Dup_I **(d)** Dup_{II}

Figure 6.8: *Cumulative Match Curves for the vectors y_T and y_F using Mag, LoG and γ on the four FERET probe sets (a)f_b (b) f_c, (c) Dup_I and (d)Dup_{II}*

Method	FERET Probe Sets			
	$\mathbf{f_b}$	$\mathbf{f_c}$	$\mathbf{Dup_I}$	$\mathbf{Dup_{II}}$
Best Results (Phillips *et al.*, 2000)	0.96	0.82	0.59	0.52
LGBPHS_Weighted (Zhang *et al.*, 2005)	0.98	0.97	0.74	0.71
HGPP_Weighted (Zhang *et al.*, 2007)	0.97	0.99	0.80	0.78
Gabor+LBP_KDCVM (Tan and Triggs, 2007)	0.98	0.98	0.90	0.85
GRM-Local (Lui and Beveridge, 2008)	0.98	0.98	0.80	0.84
$y_F(Mag, Log, \gamma)$ + **KDCV** ($q = 28322$)	**0.98**	**0.94**	**0.88**	**0.82**
$y_T(Mag, Log, \gamma)$ + **KDCV** ($q = 18818$)	**0.99**	**0.99**	**0.91**	**0.86**

Table 6.5: *The Rank-1 Recognition Rates of Different Algorithms on the FERET Probe Sets*

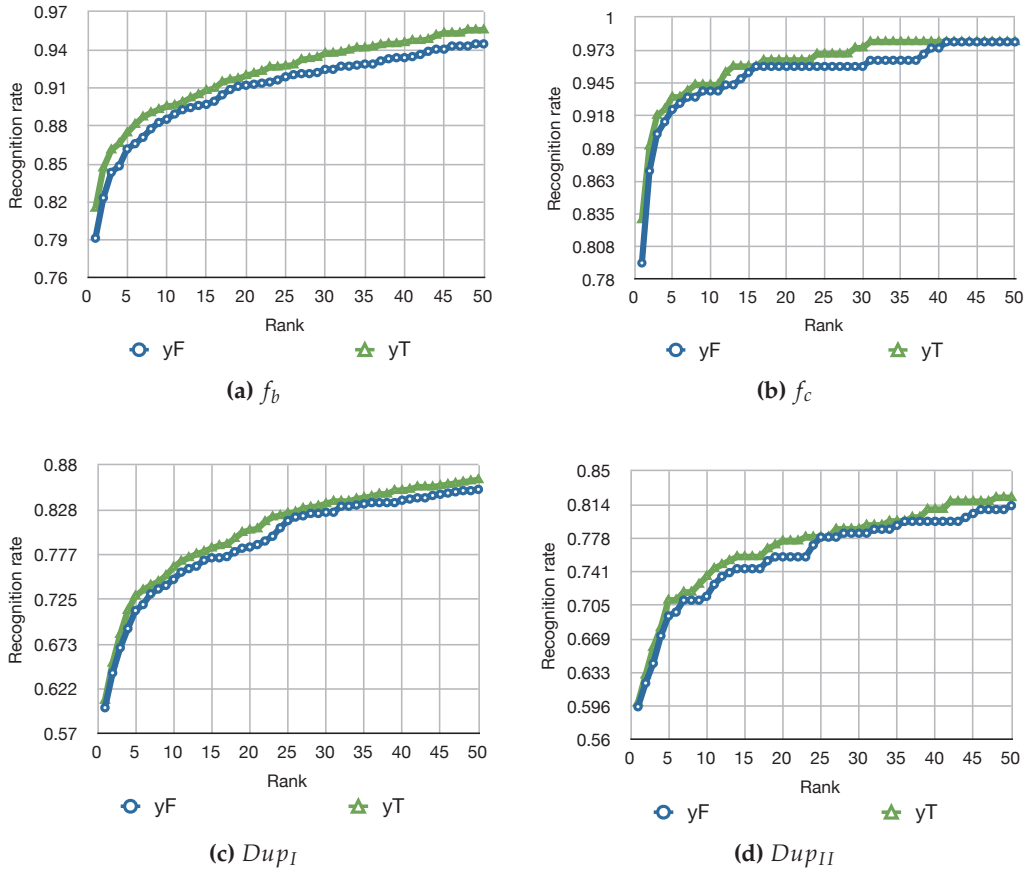


Figure 6.9: *Rank-1 recognition accuracy versus dimensionality reduction with MPCA for the vectors y_T and y_F using Mag, LoG and γ on the four FERET probe sets (a)f_b (b) f_c, (c) Dup_I and (d)Dup_{II}*

the results in table 6.6. The best two results for a particular training set size are highlighted in bold and compared with the learning-based face recognition methods. As far as we know, the results in Kumanr *et al.* (2009) are the most recent state-of-the-art results in the Yale + extended Yale dataset.

6.6 Summary and Conclusion

We have introduced a new method for recognizing faces that uses a simple set of Gaussian maps and a tensorial representation that conserves the natural structure of the feature space. Two algorithmic structures for face recognition system have been compared one using the concatenated vector y_F and other using y_T, in both cases the features used are shown an improvement in performance for an specific case as illumination or age variations, besides with the tensorial representation and MPCA is possible to generate an only vector that gather such performances. We have discussed the advantages and disadvantages of each step in these two approaches and have conducted a set of experiments to show the utility of each

component in the algorithm, in the experimental scope, we observe a clearly improvement of recognition rate when y_T is used.

Our proposed tensorial representation using Gaussian maps is competitive with state-of-the-art methods that use similar techniques with more complicated features such as Gabor maps Zhang *et al.* (2005, 2007); Tan and Triggs (2007); Lui and Beveridge (2008) in all the probe sets from the FERET dataset. We have also shown that neither face recognition algorithm needs correction in illumination to outperforms other methods Hua *et al.* (2007); Cai *et al.* (2007); An *et al.* (2008); Fu and Huang (2008); Kumanr *et al.* (2009) in the challenging Yale B + Yale extended data set and finally we shown a certain invariance to expression in the results reported using the Yale data set.

	Train Set Size			
Method	5	10	20	30
ORO (Hua *et al.*, 2007)	-	-	-	9.0
SR (Cai *et al.*, 2007)	-	12.0	4.7	2.0
RDA (Cai *et al.*, 2007)	-	11.6	4.2	1.8
KLPPSI (An *et al.*, 2008)	24.74	9.93	3.15	1.39
CTA (Fu and Huang, 2008)	16.99	7.60	4.96	2.94
Eigenfaces (Fu and Huang, 2008)	54.73	36.06	31.22	27.71
Fisherfaces (Fu and Huang, 2008)	37.56	18.91	16.87	14.94
Laplacianfaces (Fu and Huang, 2008)	34.08	18.03	30.26	20.20
Volterrafaces (Linear) (Kumanr *et al.*, 2009)	6.35	2.67	0.90	0.42
Volterrafaces (Quad) (Kumanr *et al.*, 2009)	13.0	3.98	1.27	0.58
$y_F(Mag, Log, \gamma)$ **+ KDCV** $(q = 28322)$	6.94	**1.12**	**0.36**	**0.33**
$y_T(Mag, Log, \gamma)$ **+ KDCV** $(q = 18818)$	**6.61**	**0.90**	**0.32**	**0.32**

	Train Set Size			
Method	2	3	4	40
MLASSO (Pham and Venkatesh, 2008)	58.0	54.0	50	-
SR (Cai *et al.*, 2007)	-	-	-	1.0
RDA (Cai *et al.*, 2007)	-	-	-	0.9
Volterrafaces (Linear) (Kumanr *et al.*, 2009)	26.23	18.23	9.33	0.34
Volterrafaces (Quad) (Kumanr *et al.*, 2009)	40.81	20.47	14.42	0.43
$y_F(Mag, Log, \gamma)$ **+ KDCV** $(q = 28322)$	**32.18**	**16.46**	10.08	**0.17**
$y_T(Mag, Log, \gamma)$ **+ KDCV** $(q = 18818)$	32.42	**16.26**	**9.73**	**0.19**

Table 6.6: *The Rank-1 average recognition error rates on the Yale B + Extended Yale B dataset with different training set sizes*

Estimation de l'âge avec les HBGM

Étend l'application des représentations tensorielles au problème de l'estimation de l'âge en utilisant les caractéristiques des dérivés gaussien. En particulier, ce chapitre aborde le problème de l'estimation de l'âge comme un problème de régression en utilisant les vecteurs y_T et y_F (voir la section 7.4) comment entrées pour la modélisation d'un régresseur en utilisant des Machines a vecteurs de pertinence (RVM). Pour résoudre le problème de l'estimation de l'âge du visage, nous utilisons des dérivés gaussiens jusqu'à le quatrième ordre pour obtenir des important caractéristiques faciales qui décrivent le processus de vieillissement et qui ne peuvent pas être décrits en utilisant uniquement des dérivés gaussiens d'ordre inférieur.

Deux ensembles de données disponibles au public (FG-net et MORPH) sont utilisés pour montrer la qualité de l'approche pour résoudre ce problème. Les résultats sont compétitifs avec le dernier état de l'art des méthodes proposées dans le domaine de l'analyse faciale (voir les sections 7.5 et 7.6).

7 Age Estimation using HBGM

Chapter Contents

7.1 Motivation **103**
7.2 Theoretical Background **104**
7.2.1 Relevance Vector Machines (RVM) as regressor 104
7.3 Age estimation using RVM **105**
7.3.1 Experimental Datasets 106
7.4 Comparing y_T and y_F performances **107**
7.5 Results in the FG-NET database **108**
7.6 Results in the MORPH(test) Database **110**
7.7 Summary and Conclusion **111**

7.1 Motivation

As described in section 2.3, faces are an important source of information which has been considered in Human-Computer Interfaces (HCI), law-enforcement applications and video surveillance. Between all this information, we can find the aging information. Facial age estimation is a complicated problem which has been highly studied by the computer vision community. Solutions have been proposed using advanced machine learning algorithms, Active Appearance Models and template matching. In this chapter age estimation problem has been addressed using Histograms of Binary Gaussian Maps (HBGM) representation explained in chapter 5 and Relevance Vector Machines as regressors. In age estimation from faces, HBGM provides a robust facial representation capable of encoding aging information in two ways: in appearance using Binary Gaussian Maps and in shape using tensorial representations, both of them combined using Multilinear principal Component Analysis, provide a robust facial feature.

7.2 Theoretical Background

7.2.1 Relevance Vector Machines (RVM) as regressor

The relevance vector machine has been proposed by Tipping (2001) to adapt the main ideas of Support Vector Machines (SVM) to a Bayesian context. Experimental results using RVM as regression have been shown to be as accurate and sparse as SVMs. The main advantage of RVMs is that they do not require a complicated set-up of free parameters found in SVM. Such set-up requires a long cross validation process or using kernel optimizations as it is described by Lampert (2009).

The objective of RVMs as regressors is to learn a vector of weights $\mathbf{w}$ which allows a mapping between an input vector $\mathbf{x}$ and output training target t_i defined as follows:

$$t_i = \mathbf{w}^T \phi(\mathbf{x}_i) + \varepsilon_i \tag{7.1}$$

where $\phi(x)$ corresponds to a kernel function that maps x_i in a non-linear subspace and ε_i corresponds to a gaussian noise with zero mean and variance σ^2.

For finding the weights $\boldsymbol{w}$, The RVM algorithms uses the Gaussian prior $P(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=0}^{N} \mathcal{N}\left(0, \alpha_i^{-1}\right)$ where α_i describes the inverse variance (or relevance) of each w_i and N is the number of training samples. Using the premise above mentioned, it is possible to express the posterior probability over all the unknown parameters as follows:

$$P\left(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2 \mid \boldsymbol{t}\right) = P\left(\boldsymbol{w} \mid \boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2\right) P\left(\boldsymbol{\alpha}, \sigma^2 \mid \boldsymbol{t}\right) \tag{7.2}$$

where $P\left(\boldsymbol{w} \mid \boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2\right) \backsim \mathcal{N}\left(\boldsymbol{m}, \boldsymbol{\Sigma}\right)$ with mean $\mathbf{m} = \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^T$, covariance $\boldsymbol{\Sigma} = \left(\mathbf{A} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1}$, $\boldsymbol{A} = diag\left(\boldsymbol{\alpha}\right)$ and $\beta^{-1} = \sigma^2$.

Finally, to evaluate $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$, it is necessary to find the hyper-parameters α_i and β, for doing this, we need to solve the next log-marginal likelihood which has been developed using the second part from the precedent equation.

$$\ln P\left(\boldsymbol{t} \mid \boldsymbol{\alpha}, \beta\right) = \frac{N}{2} \ln \beta - \frac{1}{2}\left(\beta \boldsymbol{t}^T \boldsymbol{t} - \boldsymbol{m}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{m}\right) - \frac{1}{2} \ln |\Sigma| - \frac{N}{2} \ln\left(2\pi\right) + \frac{1}{2} \sum_{i=1}^{N} \ln \alpha_i \tag{7.3}$$

This likelihood can be maximized using the the evidence approximation procedure presented in Tipping (2001), giving the next solutions:

$$\alpha_i = \frac{\gamma_i}{m_i^2} \tag{7.4}$$

$$\beta = \frac{N - \Sigma_i \gamma_i}{\left|\boldsymbol{t} - \boldsymbol{\Phi} \boldsymbol{m}\right|^2} \tag{7.5}$$

The algorithm 6, summarizes the training procedure for the Relevance Vector Machines.

Algorithm 6 Relevance Vector Machines (Tipping, 2001). (courtesy of Fletcher (2010))

{Giving a set of N training vectors $\mathbf{x}$, and a vector $\mathbf{t}$ represents all the individual training points t_i corresponding to x_i}.
{Select a suitable kernel function ϕ for the data set and relevant parameters. Use this kernel function to create the design matrix $\mathbf{\Phi}$ }
{Establish a suitable convergence criteria for $\boldsymbol{\alpha}$ and β, e.g. a threshold value for change δ_{Thresh} between one iteration's estimation of $\boldsymbol{\alpha}$ and the next $\delta = \sum_{i=1} \left(\alpha_i^{n+1} - \alpha_i^n\right)$}
{Establish a threshold value α_{Thresh} which it is assumed an α_i is tending ton infinity upon reaching it}
{Choose starting values for $\boldsymbol{\alpha}$ and β}
while $\delta > \delta_{Thresh}$ **do**
  Calculate $\mathbf{m} = \beta\mathbf{\Sigma}\mathbf{\Phi}^T$ and $\mathbf{\Sigma} = \left(\mathbf{A} + \beta\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}$
  Update $\alpha_i = \frac{\gamma_i}{m_i^2}$ and $\beta = \frac{N - \Sigma_i \gamma_i}{\|\mathbf{t} - \mathbf{\Phi}\mathbf{m}\|^2}$
  Prune the α_i and corresponding basis functions where $\alpha_i > \alpha_{Thresh}$
The output are the hyper-parameters $\boldsymbol{\alpha}$,β and $\mathbf{m}$

7.3 Age estimation using RVM

To estimate age from faces, we have used the architectures shown in figure 7.1. In both cases the resulting vector, after applying MPCA, was used as input to train a regressor. When a candidate facial image is present, first we apply our tensorial transformation mentioned in chapter 5, then once the final vectors are computed, we try to determine its correct age using the learned regression function. In the follow sections, we going to show that our tensorial representations can encode the necessary facial information for determining the age of a subject using the trained regression function. In all of our experiments, the images were cropped using manually located eye positions and normalized in size to 64×64 pixels. The Binary Gaussian Receptive Maps are calculated in a half-octave Gaussian pyramid with four levels ($\sigma = \sqrt{2}, 2, 2\sqrt{2}$ and 4). A border of 4 pixels in each pyramid level is left untested for faces to avoid problems related with image borders.

To calculate histograms, we used a sub-region size of 8×8 pixels, removing two bins corresponding to values of 0 and 255 per each histogram. The remaining bins are grouped to form a 127 bin histogram. We used the publicly available MATLAB® implementation of the RVM algorithm provided by Tipping (2001)[1]. After applying MPCA, each final vector is normalized to unit standard deviation. For the RVM algorithm we used a Gaussian kernel $k(x, y) = e^{-\|x-y\|^2/q}$ with a scale

[1] http://www.vectoranomaly.com/downloads/downloads.htm

parameter q determined using a tuning dataset chosen randomly from the training dataset (the value of q is reported in each experiment and is the only one fixed by tuning).

The performance of age estimation is measured by the mean absolute error (MAE) and the cumulative score (CS).

- The MAE is defined as the average of the absolute errors between the estimated ages and the ground truth ages $MAE = \sum_{k=1}^{N} |\hat{Age}_k - Age_k|/N$, where Age_k is the ground truth age for the test image k, $\hat{Age}_k$ is the estimated age, and N is the total number of test images. MAE is only an indicator of average performance for age estimators, it does not provide enough information of how accurate the estimators might be.

- The accuracy can be estimated by the cumulative score (CS) that is defined as $CS\,(j) = \frac{N_{e \leq j}}{N} \times 100$, where $N_{e \leq j}$ is the number of test images on which the estimator makes an absolute error non-higher than j years.

7.3.1 Experimental Datasets

We have performed several experiments to compare different approaches for estimating age from facial images. Two publicly available databases have been used in our experiments: The FG-NET database [2] and the MORPH (Ricanek and Tesafaye, 2006) database. Some examples of images from these datasets are shown in figures 7.2 and 7.3.

7.3.1.1 FG-NET Aging Dataset

The FG-NET (Face and Gesture Recognition Research) Aging Database contains 1,002 face images of 82 subjects from multiple races with age ranges from 0 to 69 years. Each image in the database has 68 labeled facial landmarks characterizing shape features not used in our approach. Since the images were retrieved from real-life albums of different subjects, aspects as illumination, head pose, facial expressions etc. are uncontrolled in this dataset.

Leave-One-Person-Out (LOPO) mode is used for testing our approaches in the FG-NET database. In this mode, the images of one person are used as the test set and those of the others are used as the training set. After 82 folds each subject has been used as a the test set once and the final results are calculated based in the result of each fold.

7.3.1.2 MORPH Aging Dataset

The MORPH Aging Database contains 1,724 face images of 515 subjects. In our experiment with this database, we use the same testing protocol used by Geng *et al.*

[2] The FG-NET aging database, `http://www.fgnet.rsunit.com/`, Accessed: April 2010

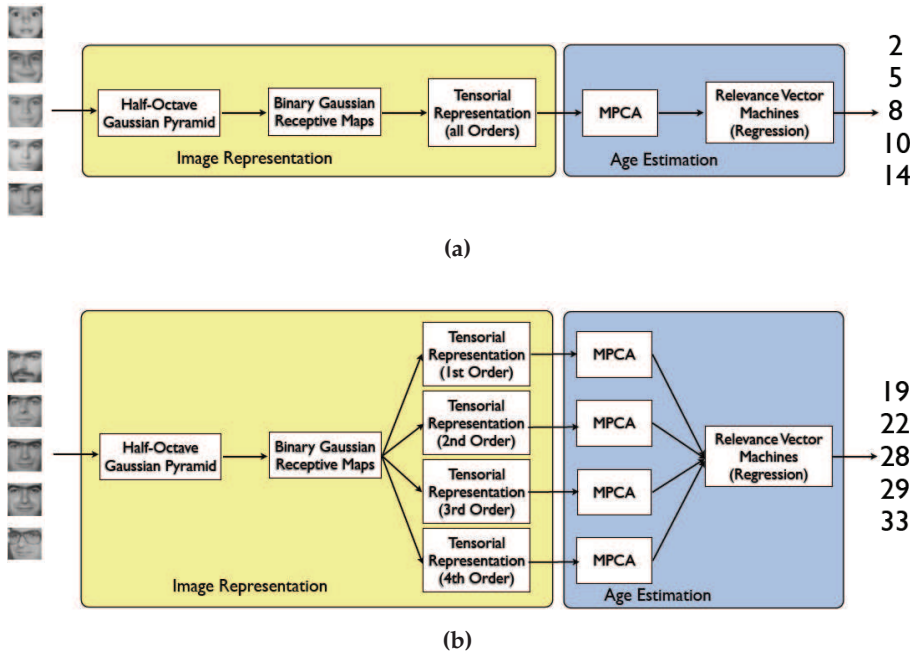


Figure 7.1: *Age estimation architectures proposed in this thesis using tensors of HBGM. (a)concatenated vectors(y_T) then RVM. (b) concatenated tensors then MPCA (y_F)and finally RVM*

(2007). The images on these dataset are only used to test the algorithms trained on the FG-NET database. In addition, because all subjects in the FG-NET database are Caucasian descent, only the 433 images of Caucasian descent in the MORPH database are used as the test set.

7.4 Comparing y_T and y_F performances

Our first experiment investigates the performance of each type of configuration for the age estimation problem. We compared the configurations showed in figure 7.1 on the FG-NET database and we report the results in table 7.1. Different orientations and Gaussian derivative orders were tested to obtain the best configuration.

Figure 7.2: *Sample images of different individuals from the FG-NET dataset*

Figure 7.3: *Sample images of different individuals from the MORPH dataset*

We used orientations between 0 and π and derivative orders up to fourth, giving the next bank of Gaussian derivative filters used in our age estimation experiments for computing the Histograms of Gaussian Binary Maps (HGBM).

$$\mathbf{F}_{(n,\theta\in[0,\pi])} = G_{n,\theta}\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}\right) \quad with \quad n = \{1, 2, 3, 4\} \tag{7.6}$$

Using this filter bank, we are able to compute four diferent tensors:

$$\mathcal{T}_n \in \mathbb{R}^{32\times orientations\times 127} \quad n = \{1, 2, 3, 4\} \tag{7.7}$$

Experimental results show that the best performance can be achieved with 8 orientations $(0, \pi/7, 2\pi/7, 3\pi/7, 4\pi/7, 5\pi/7, 6\pi/7\ and\ \pi)$ and four derivative orders, organized in a y_F configuration, the second best result was achieved with 6 orientations $(0, \pi/5, 2\pi/5, 3\pi/5, 4\pi/5\ and\ \pi)$ and three derivative orders in a y_T configuration. The best results were highlighted in the table 7.1 and used in the following experiments.

7.5 Results in the FG-NET database

We compared the two best tensorial configurations of table 7.1 with the most relevant results of the state-of-the-art in age estimation. In table 7.2, we report the results for seven different age groups between 0 and 69 years. In this table we observed that our method outperforms other methods for age groups 40 - 59 and 20-29 years. For other age groups, our method is competitive and sometimes superior with competing approaches in age estimation. The experiments the robustness of our method to textural changes that occur in the periods from adulthood to old age.

In the FG-NET database, the MAEs of our method are 5.16 for the y_T configuration and 4.96 for the y_F configuration. Comparisons with alternative approaches are reported in table 7.3. Our results are comparable with the latest state-of-the-art methods in automatic age estimation in the FG-NET database.

Comparisons of cumulative scores (CS) on the FG-NET database are shown in Figure 7.4a. We can observe that despite the MAE results for our methods, our approaches outperforms state-of-the-art methods in low age error levels ($Error\ level\ \leq\ 4$), with almost 5% of improvement in accuracy $CS_{\leq 4} = 73\%$, in

	y_T + RVM		
	$n = \{1,2,3\}$	$n = \{1,2,3,4\}$	$n = \{3,4\}$
Orientations			
$\theta = \{0 : \frac{\pi}{5} : \pi\}$	5.25 ($q = 28.72$)	5.25 ($q = 29.58$)	5.53 ($q = 29.58$)
$\theta = \{0 : \frac{\pi}{7} : \pi\}$	**5.16 ($q = 30.82$)**	5.23 ($q = 30.82$)	5.48 ($q = 30.82$)
	y_F + RVM		
	$n = \{1,2,3\}$	$n = \{1,2,3,4\}$	$n = \{3,4\}$
Orientations			
$\theta = \{0 : \frac{\pi}{5} : \pi\}$	5.23 ($q = 50.00$)	5.17 ($q = 66.33$)	5.49 ($q = 38.73$)
$\theta = \{0 : \frac{\pi}{7} : \pi\}$	5.18 ($q = 50.00$)	**4.96 ($q = 66.33$)**	5.46 ($q = 38.73$)

Table 7.1: *MAEs on the FG-NET database for different tensorial configurations*

		Method		
Range	**# img.**	y_T **+ RVM**	y_F **+ RVM**	BIF (Guo *et al.*, 2009)
0-9	371	3.14	3.19	2.99
10-19	339	4.05	3.90	3.39
20-29	144	4.72	4.29	4.30
30-39	70	10.08	9.17	8.24
40-49	46	14.15	13.76	14.98
50-59	15	22.06	20.06	20.49
60-69	8	33.12	32.25	31.62
Total	1002	**5.16**	**4.96**	4.77

		Method		
Range	**# img.**	RUN (Yan *et al.*, 2007a)	QM (Lanitis. *et al.*, 2004)	MLP (Lanitis. *et al.*, 2004)
0-9	371	2.51	6.26	11.63
10-19	339	3.76	5.85	3.33
20-29	144	6.38	7.10	8.81
30-39	70	12.51	11.56	18.46
40-49	46	20.09	14.80	27.98
50-59	15	28.07	24.27	49.13
60-69	8	42.50	37.38	49.13
Total	1002	5.78	7.57	10.39

Table 7.2: *MAE (years) at different age groups on FG-NET.*

Method	MAE(Years)
QM (Lanitis. *et al.*, 2004)	6.55
MLPs (Lanitis. *et al.*, 2004)	6.98
RUN (Yan *et al.*, 2007a)	5.78
BM Yan *et al.* (2008b)	5.33
LARR (Guo *et al.*, 2008a)	5.07
PFA (Guo *et al.*, 2008b)	4.97
BIF Guo *et al.* (2009)	4.77
y_T**+RVM**	**5.16**
y_F**+RVM**	**4.96**

Table 7.3: *MAE (years) comparisons on FG-NET*

addition for high error levels our method has an $CS_{\leq 10} = 88\%$ similar to BIF Guo *et al.* (2009) ($CS_{\leq 10} = 89\%$) that as far as we know the best result in the FG-NET dataset.

7.6 Results in the MORPH(test) Database

More experiments were conducted on the MORPH(test) aging database. From the results in the table 7.4, the MAEs results of our method are 6.19 and 6.76 for y_F and y_T respectively, those results outperforms the AGES method Geng *et al.* (2007) in almost two years of difference and other methods like SVM and WAS with a difference of almost three years.

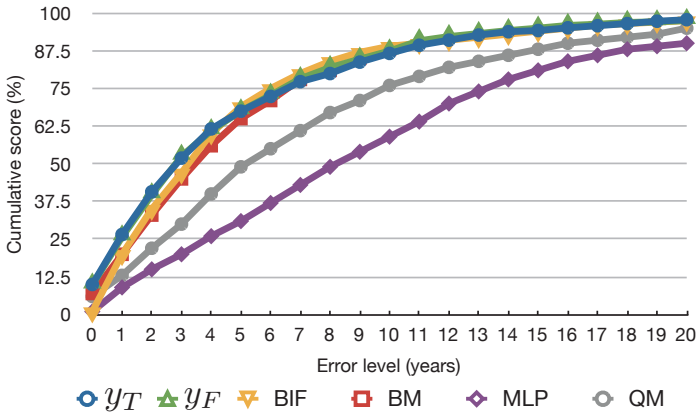
The CS curves on the MORPH database are shown in Figure 7.4b. Our method outperforms other methods in error levels for all of the age groups with an $CS_{\leq 10} = 84\%$ against the AGES*lda* method Geng *et al.* (2007) with a $CS_{\leq 10} = 78\%$.

Method	MAE(Years)
WAS (Geng *et al.*, 2007)	9.32
SVM (Geng *et al.*, 2007)	9.23
AGES (Geng *et al.*, 2007)	8.83
AGES*lda* (Geng *et al.*, 2007)	9.32
y_T**+RVM**	**6.77**
y_F**+RVM**	**6.19**

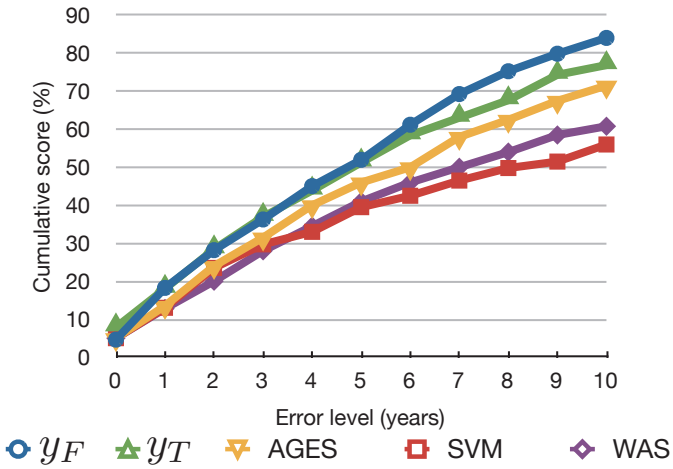
Table 7.4: *MAE (years) comparisons on MORPH (Test Set)*

7.7 Summary and Conclusion

In this chapter, we have introduced a method to recognize the age through facial images. This method uses HBGM as image representation and Relevance Vector Machines as regressor. For finding the most adapted configuration for the age estimation task, different derivative orders were used in our two proposed tensorial representations. In this point our results have shown that using Gaussian derivatives up of the fourth order improve the results considerably in both configurations. In other hand we have shown experimentally than our tensorial configurations are suited for modeling facial age information which is used in the regression process. Besides the main problem encountered in facial age estimation using regresion methods is the difficulty in constructing a suitable set of training images whose represent the age progression in different subjects. Finally, we have compared our method with previous state-of-the-art approaches in aging estimation.



(a) *Cumulative scores on the FG-NET database*



(b) *Cumulative scores on the MORPH (test) database*

Figure 7.4: *Cumulative scores on (a)FG-NET database and (b)MORPH (test) database*

Conclusions et perspectives

Conclut les principaux résultats et liste les perspectives de la thèse.

8 Conclusion and perspective

Chapter Contents

8.1 Principal results 115
8.2 Perspectives 116

In this thesis we have explored the use of multiscale Gaussian derivative as an initial representation for the detection, recognition and classification of faces in images. The results of our investigation show that multiscale Gaussian derivatives can provide a rich image description that robustly capture the appearance of faces in images, supporting a variety of different techniques for detection and recognition. In particular, we have shown that this representation can be used to obtain detection and recognition rates that are comparable to the best state of the art algorithms with significant reductions in computational cost and a computational structure that is suitable for embedded applications on mobile devices.

8.1 Principal results

Among the principal results of this investigation, we can list:

- Multiscale Gaussian derivatives computed at half-octave scale intervals can provide a powerful intial representation for detection, recognition and classification.
- Including Gaussian derivatives features of up to fourth order can be used to improve detection results with real-world images..
- A Tensorial representation can be constructed from multiscale derivatives and used for both Face Recognition and Age estimation.

- A half-octave pyramid representation can be used to construct an efficient variable density sampling algorithm for fast face detection.

Throughout this thesis we have been careful to provide the mathematical and theoretical justification for our approach.

In chapter 4, we proposed the use of a cascade of simple classifiers using Gaussian derivatives features up to fourth order. We show that Gaussian derivatives features are suitable for face detection in real-world images specially those were the detection precision is fundamental, this idea was confirmed by the results in the FDDB and MIT+CMU datasets.

In addition a new speed-optimized cascade framework was proposed. This new framework takes into account the computational load of each Gaussian derivatives to chose their position in an specific node in the cascade. Our experiments using this framework have shown that using Gaussian derivatives features of first and second order in the first layers and higher order for deeper nodes improves the speed of the cascade in almost twice compared with the classical approach using all derivative orders in the first node.

A set of experiments are presented to show the robustness of Gaussian derivatives to different image conditions present in real-world applications, in this case Gaussian derivatives features variations in rotation, contrast and lighting.

In chapter 5, we introduce a new tensorial model using Gaussian derivative maps, Local Binary Patterns(LBP) and Kernel Discriminative common Vectors (KDCV). Two different tensorial representations were explored and all the mathematical background in reduction dimensions for multilinear representations was explained (Multilinear Principal Component Analysis).

In chapter 6, we have explored Face Recognition as a first application for the proposed tensorial representation. As principal result Tensorial models using Gaussian derivatives provides a robust representation invariant to illumination conditions as we observe in the results with the Yale B + Extended Yale B dataset. Besides, results using the Feret dataset showed that our tensorial representation is competitive with state-of-the-art approaches in the challenging Feret dataset. Finally, In chapter 7, age estimation was also studied as an application in tensorial representations, In this scope, tensorial representation encodes facial age information using Gaussian derivatives up to fourth order. Although the complexity of the age estimation problem, our results in the FG-NET and MORPH datasets have showed that our proposed tensorial representation combined with Relevance Vector Machines is a reliable solution that could be implemented in real world applications without a high computational cost.

8.2 Perspectives

Several interesting questions remains open, some of which provide potentially interesting pathways for future work. These are discussed in the following section:

- **Can multiscale Gaussian derivatives be used for action unit detection for Emotion Recognition from faces**. Initial experiments suggest that Gaussian derivatives can be used to construct a new kind of feature based in first and second order derivatives mixed in a complex conjugate form $G_1 + G_2 j$. First experiments with this approach appear promising, we are about to initiate a more complete set of experiments. A related question concerns the contribution that higher order derivatives can bring to emotion recognition from facial action unit detection.

- **Gender Recognition** as an extension of the cascade of classifiers, Gender recognition can be solved using Gaussian derivatives in a cascade framework. Our future aim is to train some extra-nodes after the face detection process. Such nodes will be trained to recognize gender (male or female). In addition, it could be interesting to analyze more deeply which or whose derivative order could perform the best in gender recognition where finest facial details are necessary to distinguish between genders.

- **Face Recognition across ages**, in this thesis face recognition was considered as a problem where the age of the people is not an important variable. Moreover, face recognition across ages remains an important problem in facial analysis. In this scope, aging information could be introduced as a new dimension in the tensorial representation. This new dimension in conjunction with MPCA will retain the most common facial features that could be invariant of transformation due to age.

- **Statistical learning methods for tensorial representations** is the special interest, in effect extracting the most discriminative information from a tensor could be advantageous when the feature space is composed of a big number of dimensions. On the other hand, when some information is missing it is not possible to construct a discriminative tensorial representation (i.e. age estimation datasets where for a particular subject all their images are not available or missed) that can be used in multi-class learning problem. In this case, the statistical tensorial framework allows us to complete such information using the information provided by other subjects in the database. In the future, we would like to address this problem, taking into a account precedent approaches that use Multilinear Subspace Analysis with missing values (Geng and Smith-Miles, 2009; Geng *et al.*, 2011).

A

Gaussian Derivatives with the Half-Octave Gaussian Pyramid

Chapter Contents

A.1 The pyramid algorithm 120
A.2 Gaussian Derivative Feature Calculation 124
A.3 Oriented Derivatives 126
A.4 Scale Interpolated Derivative 127

View Invariant Gaussian derivative features are computed by interpolation from samples provided by a multi-resolution binomial pyramid.

The binomial pyramid calculation produces K resampled copies of the input buffer at an exponentially sequenced progression of smoothing scales, where K is based on the extracted window size, and W and H are the width and height of the window.

$$K = 2\,log_2(min\{W, H\}) \tag{A.1}$$

In practice, the final 5 pyramid levels, with sizes of 8×8 and smaller, are generally discarded because they are dominated by boundary effects. However, in our analysis below, we will consider the entire pyramid composed of K levels. The computational requirements for the last 5 levels are trivial.

Each image in the pyramid has a $\sqrt{2}$ reduction in resolution (due to smoothing) and a $\sqrt{2}$ increase in the distance between samples. The increase in sample distance exactly compensates the growth of the impulse response. As a result, the sampled impulse responses from each level of the pyramid are identical copies, providing for scale invariant feature description.

For a window size of $N = W \times H$ pixels, the K levels of the pyramid produce $P = 2N$ pixels. These pixels can be used to synthesize the Gaussian impulse

response for any position, orientation or scale over the range of positions and scales recorded in the pyramid.

A.1 The pyramid algorithm

The pyramid algorithm is a repetition of k identical SIMD computations. Each stage involves resampling with a distance of $\sqrt{2}$ followed by convolution with a Binomial filter of standard deviation σ_0. Our analysis and experiments (shown below) indicate that $\sigma_0 = 1$ provides an acceptable level of smoothing to avoid distortion from aliasing.

The effect of cascade convolution is to sum the variances of the filters, so that the cumulative variance is $\sigma_k^2 = 2^k$ and the resulting standard deviation is $\sigma_k = 2^{k/2}$. Interleaving resampling with convolutions decreases the number of image samples while expanding the distance between samples. This has the effect of dilating the Gaussian support without increasing the number of samples used for the Gaussian, effectively increasing the scale. Aliasing is avoided because the images have been low-pass filtered by previous convolutions. The result is an algorithm with linear algorithmic complexity (i.e. $O(N)$), that gives a discrete representation of scale space with $P = 2N$ total samples at a total cost of three convolutions operations per pixel. The exact cost of a convolution operation depends on how the convolution is implemented, according to a tradeoff between silicon surface and computation time. For example, convolution with the kernel filter $B_4(x, y)$ used below can be implemented as separable convolutions in the row and column directions using 2 passes in each direction (row and column) with $[1, 2, 1]$, at a total cost of 8 adds and 4 shifts per pixel, followed by a normalization implemented by shifting to the right 1 bit.

The pyramid is computed by initially convolving the image window with an integer coefficient binomial filter, followed by a repeated (cascade) computation with pipeline of pyramid stage. After the first stage, each successive stage begins by resampling the input image window to select every second column of every second row. The pyramid repeats until the resampled images are reduced to a single pixel.

Analysis, backed by experimental verification, has shown that $\sigma = 1$ provides the smallest kernel filter with insignificant aliasing. This corresponds to the Binomial filter $B_4(i, j)$.

$$B_4(i,j) = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{A.2}$$

The image is initially convolved with a filter of $\sigma_0^2 = 1$ to produce an initial image

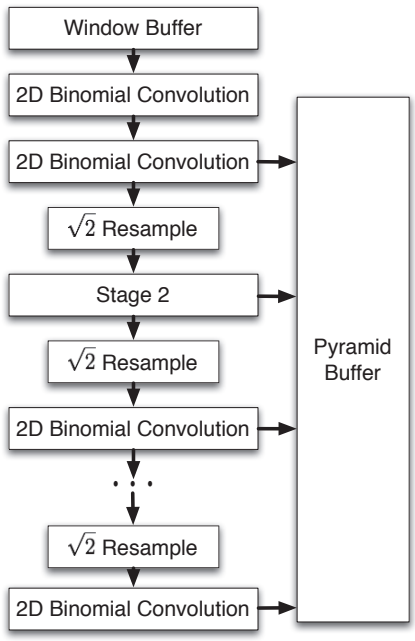


Figure A.1: *The $O(N)$ cascade convolution pyramid algorithm*

$P(x, y, 0)$.

$$k = 0 \Rightarrow P(i, j, 0) = P(i, j) * B_4(i, j) \tag{A.3}$$

where $*$ is the convolution operator. The pyramid image $(k = 1)$ is produced by a convolution with the same low pass filter, resulting in a cumulative scale factor of $\sigma_1^2 = 2$ giving $\sigma_1 = \sqrt{2}$

$$k = 1 \Rightarrow P(i, j, 1) = P(i, j, 0) * B_4(i, j) \tag{A.4}$$

For even numbered stages, diagonal sampling suppresses every second pixel starting with even columns on even rows and odd columns on odd rows, as shown in figure A.2. For k even, the operator eliminates every second row.

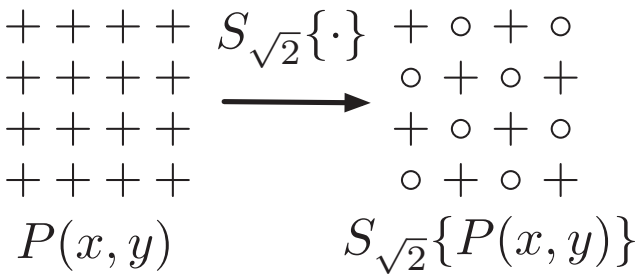


Figure A.2: *The $\sqrt{2}$ resampling Operator, $S_{\sqrt{2}}\{\cdot\}$, selects even columns of even rows and odd columns of odd rows.*

$$S_{\sqrt{2}^k}\{P(x,y)\} = \begin{cases} P(x,y) & if \quad (x+y)\, Mod\, 2^k = 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.5}$$

The resulting pixels are stored in a rectangular buffer of size $\left(\frac{W}{2^{\frac{(k-1)}{w}}}, \frac{H}{2^{\frac{k}{2}}}\right)$. The one pixel shift to the right for even rows is implicit.

For odd stages the binomial filter must be mapped onto the diagonal sample grid shown below. This is accomplished by the diagonal expansion operator, $E_{\sqrt{2}}\{\cdot\}$, shown in figure A.3.

$$E_{\sqrt{2}^k}\{B_4(i,j)\} = \begin{cases} B_4(i,j) & if \quad (i+j)^2\, Mod\, 4 = 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.6}$$

Each successive image in the pyramid is computed by convolving an expanded Gaussian with a sampled image as described by the following recurrence equation (k refers to pyramid level):

$$P(i,j,k) = \begin{cases} P(i/2, j/2, k1) * B_4(i,j) & if \quad k \text{ is odd} \\ S_{\sqrt{2}}\{P(i,j,k1)\} * E_{\sqrt{2}}\{B_4(i,j)\} & \text{otherwise} \end{cases} \tag{A.7}$$

The $k = 0$ image may be discarded or used for estimating a Laplacian image for $k = 1$ if required. Because the $k = 1$ image has been smoothed with a Binomial low-pass filter of scale $\sigma_1 = \sqrt{2}$, resampling with a sample distance of $\sqrt{2}$ will result in an aliasing of less than 1% of signal energy.

The resulting pyramid is composed of K resampled copies of the image buffer smoothed with an exponential progression of sample rates. For a buffer of N pixels, the total number of pyramid pixels is $P = N(1 + 1/2 + 1/4 + 1/8 + 1/16 + \cdots) = 2N$.

For each of the K images, the distance between pixels is:

$$S_k = 2^{(k-1)/2} \tag{A.8}$$

For odd levels, these samples sit on Cartesian positions (i,j), in a W_k by H_k where

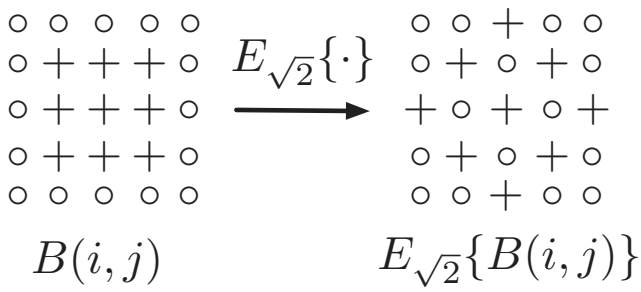


Figure A.3: *The $\sqrt{2}$ expansion operator, $E\sqrt{2}\{\cdot\}$, maps rows of a filter onto diagonals, increasing sample distance by $\sqrt{2}$, illustrated here for the 3×3 filter $B_2(i,j)$.*

$$\begin{bmatrix} + & . & + & . & + & . & + & . \\ . & . & . & . & . & . & . & . \\ + & . & + & . & + & . & + & . \\ . & . & . & . & . & . & . & . \\ + & . & + & . & + & . & + & . \\ . & . & . & . & . & . & . & . \\ + & . & + & . & + & . & + & . \\ . & . & . & . & . & . & . & . \end{bmatrix} \begin{bmatrix} + & . & . & . & + & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & + & . & . & . & + & . \\ . & . & . & . & . & . & . & . \\ + & . & . & . & + & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & + & . & . & . & + & . \\ . & . & . & . & . & . & . & . \end{bmatrix}$$

$$\begin{bmatrix} + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & + \end{bmatrix} \begin{bmatrix} + & . & + & . & + & . & + & . \\ . & + & . & + & . & + & . & + \\ + & . & + & . & + & . & + & . \\ . & + & . & + & . & + & . & + \\ + & . & + & . & + & . & + & . \\ . & + & . & + & . & + & . & + \\ + & . & + & . & + & . & + & . \\ . & + & . & + & . & + & . & + \end{bmatrix}$$

Figure A.4: *Illustration of diagonal sampling with levels k=1, 2, 3, 4 from a diagonally sampled pyramid. The symbol "+" represents image samples. Each image has half the pixels of the previous image.*

$$\begin{aligned} \text{for odd } k: \quad & W_k = \frac{W}{s_k} \\ \text{for odd } k: \quad & H_k = \frac{H}{s_k} \end{aligned} \tag{A.9}$$

For any pixel $P(i, j, k)$ in an odd level, the corresponding position in the image window from which the pyramid was constructed by the following formula :

$$\text{for odd } k: \quad x = i \cdot 2^{\frac{(k-1)}{2}} \qquad y = j \cdot 2^{\frac{(k-1)}{2}} \tag{A.10}$$

For even k, the samples are arrayed on a 2-sample grid, as shown in right side images of figure A.4. In this case, samples may be represented in a rectangular array with the same number of rows as the previous level, but half of the number of columns.

$$\begin{aligned} \text{for even } k: \quad & W_k = \frac{W_{k-1}}{2} = \frac{W}{2^{\frac{(k-2)}{2}}} \\ \text{for even } k: \quad & H_k = H_{k-1} = \frac{W}{2^{\frac{k}{2}}} \end{aligned} \tag{A.11}$$

The samples on even rows are implicitly shifted to the right by 1 column compared to odd rows. Thus for even rows

$$\begin{aligned} \text{for even } k: \quad & x = i \cdot 2^{k/2} + (i + j)\, Mod\, 2^{(k-1)/2} \\ \text{for even } k: \quad & y = j \cdot 2^{(k-1)/2} \end{aligned} \tag{A.12}$$

This can be stored in a $P = 2N$ sample data structure as shown in the figure A.5. Note that although 14 bits are needed during the cascade convolution, once the pyramid has been constructed, samples may be represented with 8 bits per color.

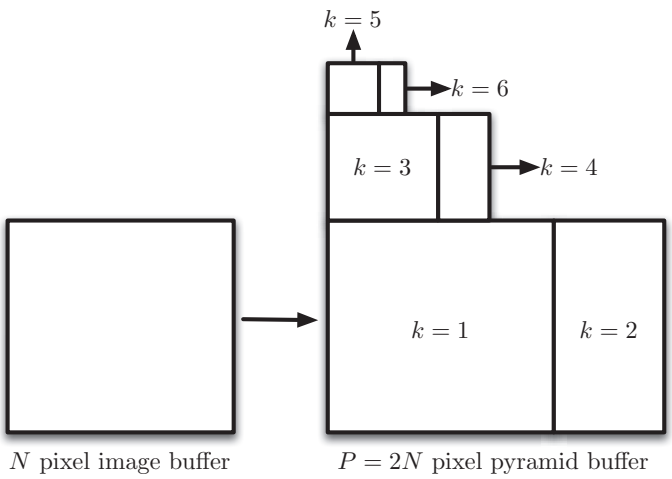


Figure A.5: *The pyramid buffer is composed of* $P = N(1 + 1/2 + 1/4 + 1/8 + \cdots) = 2N$ *samples.*

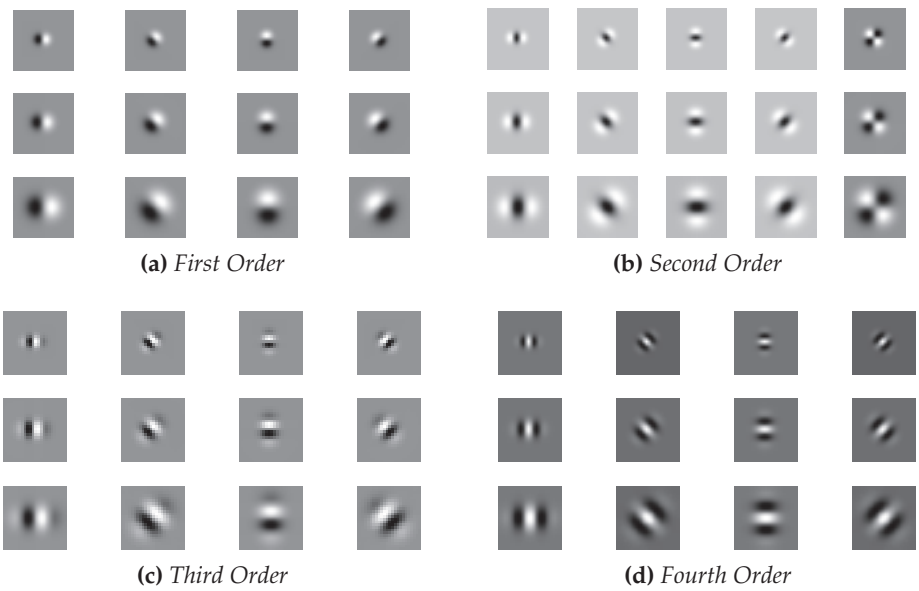


Figure A.6: *Impulse responses of Gaussian derivatives up to the fourth order computed with a Half-Octave Gaussian Pyramid*

A.2 Gaussian Derivative Feature Calculation

With Discrete Scale space, for any point i, j at level k, the derivative may be computed as a difference of adjacent samples. Because of the distance between samples and the kernel size the weighting is different for a same derivative on odd and even numbered level. Some examples of Gaussian derivatives computed with a half-octave Gaussian pyramid are shown in figure A.6.

- 1st Order derivatives

$$\begin{aligned}
G_x(i,j,k_{odd}) &= \tfrac{1}{2}\left(P(i-1,j,k_{odd}) - P(i+1,j,k_{odd})\right) \\
G_x(i,j,k_{even}) &= \tfrac{1}{2\sqrt{2}}\left(P(i-1,j,k_{even}) - P(i+1,j,k_{even})\right) \\
G_y(i,j,k_{odd}) &= \tfrac{1}{2}\left(P(i,j-1,k_{odd}) - P(i,j+1,k_{odd})\right) \\
G_y(i,j,k_{even}) &= \tfrac{1}{2\sqrt{2}}\left(P(i,j-2,k_{even}) - P(i,j-2,k_{even})\right)
\end{aligned} \tag{A.13}$$

- 2nd Order derivatives

$$\begin{aligned}
G_{x^2}(i,j,k_{odd}) &= P(i+1,j,k_{odd}) - 2 * P(i,j,k_{odd}) \\
&\quad + P(i-1,j,k_{odd}) \\
G_{x^2}(i,j,k_{even}) &= \tfrac{1}{2}\left(P(i+1,j,k_{even}) - 2P(i,j,k_{even})\right. \\
&\quad \left. + P(i-1,j,k_{even})\right) \\
G_{y^2}(i,j,k_{odd}) &= P(i,j+1,k_{odd}) - 2P(i,j,k_{odd}) \\
&\quad + P(i,j-1,k_{odd}) \\
G_{y^2}(i,j,k_{even}) &= \tfrac{1}{2}\left(P(i,j+2,k_{even}) - 2P(i,j,k_{even})\right. \\
&\quad \left. + P(i,j-2,k_{even})\right) \\
G_{xy}(i,j,k_{odd}) &= \tfrac{1}{4}\left(P(i+1,j+1,k_{odd}) - P(i+1,j-1,k_{odd})\right. \\
&\quad \left. - P(i-1,j+1,k_{odd}) + P(i-1,j-1,k_{odd})\right) \\
G_{xy}(i,j,k_{even}) &= \tfrac{1}{8}\left(P(i+1,j+2,k_{even}) - P(i+1,j-2,k_{even})\right. \\
&\quad \left. - P(i-1,j+2,k_{even}) + P(i-1,j-2,k_{even})\right)
\end{aligned} \tag{A.14}$$

- 3rd Order derivatives

$$\begin{aligned}
G_{x^3}(i,j,k_{odd}) &= \tfrac{1}{2}\left(P(i-2,j,k_{odd}) - 2P(i-1,j,k_{odd})\right. \\
&\quad \left. + 2P(i+1,j,k_{odd}) - P(i+2,j,k_{odd})\right) \\
G_{x^3}(i,j,k_{even}) &= \tfrac{1}{4\sqrt{2}}\left(P(i-2,j,k_{even}) - 2P(i-1,j,k_{even})\right. \\
&\quad \left. + 2P(i+1,j,k_{even}) - P(i+2,j,k_{even})\right) \\
G_{y^3}(i,j,k_{odd}) &= \tfrac{1}{2}\left(P(i,j+2,k_{odd}) - 2P(i,j+1,k_{odd})\right. \\
&\quad \left. + 2P(i,j-1,k_{odd}) - P(i,j-2,k_{odd})\right) \\
G_{y^3}(i,j,k_{even}) &= \tfrac{1}{4\sqrt{2}}\left(P(i,j+4,k_{even}) - 2P(i,j+2,k_{even})\right. \\
&\quad \left. + 2P(i,j-2,k_{even}) - P(i,j-4,k_{even})\right)
\end{aligned} \tag{A.15}$$

- 4th Order derivatives

$$
\begin{aligned}
G_{x^4}(i,j,k_{odd}) &= P(i-2,j,k_{odd}) - 4P(i-1,j,k_{odd}) \\
&\quad +6P(i,j,k_{odd}) - 4P(i+1,j,k_{odd}) \\
&\quad +P(i+2,j,k_{odd}) \\
G_{x^4}(i,j,k_{even}) &= \tfrac{1}{4}\left(P(i-2,j,k_{even}) - 4P(i-1,j,k_{even})\right. \\
&\quad +6P(i,j,k_{even}) - 4P(i+1,j,k_{even}) \\
&\quad \left.+P(i+2,j,k_{even})\right) \\
G_{y_4}(i,j,k_{odd}) &= P(i,j-2,k_{odd}) - 4P(i,j-1,k_{odd}) \\
&\quad +6P(i,j,k_{odd}) - 4P(i,j+1,k_{odd}) \\
&\quad +P(i,j+2,k_{odd}) \\
G_{y^4}(i,j,k_{even}) &= \tfrac{1}{8}\left(P(i,j-4,k_{even}) - 4P(i,j-2,k_{even})\right. \\
&\quad +6P(i,j,k_{even}) - 4P(i,j+2,k_{even}) \\
&\quad \left.+P(i,j+4,k_{even})\right)
\end{aligned}
\tag{A.16}
$$

A.3 Oriented Derivatives

Derivatives along diagonal direction $\left(\theta = \frac{\pi}{4}\right)$ can be computed using diagonal diferences without extra computational costs since sample are stored in memory.

- 1st derivative at $\frac{\pi}{4}$

$$
\begin{aligned}
G_{1,\frac{\pi}{4}}(i,j,k_{odd}) &= \tfrac{1}{2\sqrt{2}}\left(P(i-1,j,k+1_{odd}) - P(i+1,j-1,k_{odd})\right) \\
G_{1,\frac{\pi}{4}}(i,j,k_{even}) &= \tfrac{1}{2}\left(P(i+(j\&1)-1,j+1,k_{even})\right. \\
&\quad \left.-P(i+(j\&1),j-1,k_{even})\right)
\end{aligned}
\tag{A.17}
$$

- 2nd derivative at $\frac{\pi}{4}$

$$
\begin{aligned}
G_{2,\frac{\pi}{4}}(i,j,k_{odd}) &= \tfrac{1}{2}\left(P(i-1,j+1,k_{odd}) - 2P(i,j,k_{odd})\right. \\
&\quad \left.+P(i+1,j-1,k_{odd})\right) \\
G_{2,\frac{\pi}{4}}(i,j,k_{even}) &= P(i+(j\&1)-1,j+1,k_{even}) - 2P(i,j,k_{even}) \\
&\quad +P(i+(j\&1),j-1,k_{even})
\end{aligned}
\tag{A.18}
$$

Where:

$$
(j\&1) = \begin{cases} j+1 & \text{if } j \text{ is even} \\ 0 & \text{otherwise} \end{cases}
\tag{A.19}
$$

A.4 Scale Interpolated Derivative

For arbitrary positions on the original image (x, y) and arbitrary scale s, derivative values may be computed by interpolating between derivative values at samples. Gaussian derivatives for scale values between powers of $\sqrt{2}$ can be computed using linear interpolation between adjacent levels in the pyramid.

Bibliography

Abate, A.F., Nappi, M., Riccio, D. and Sabatino, G. "2d and 3d face recognition: A survey". In *Pattern Recognition*, 1(28):1885–1906 (2007).

Agui, T., Kokubo, Y., Nagashashi, H. and Nagao, T. "Extraction of facerecognition from monochromatic photographs using neural networks". In "Second international Conference on Automation, Robotics and Computer Vision", volume 1, pages 18.8.1–18.8.5 (1992).

Ahonen, T., Hadid, A. and Pietikainen, M. "Face description with local binary patterns: Application to face recognition". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041 (2006).

Amit, Y., Geman, D. and Jedynak, B. "Efficient focusing and face detection". In "FACE RECOGNITION: FROM THEORY TO APPLICATIONS", pages 143–158 (1998).

An, S., Liu, W. and Venkatesh, S. "Exploiting side information in locality preserving projection". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (2008).

Ashraf, A., Lucey, S. and Chen, T. "Learning patch correspondences for improved viewpoint invariant face recognition". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 1 –8 (2008).

Brubaker, S.C., Wu, J., Sun, J., Mullin, M.D. and Rehg, J.M. "On the design of cascades of boosted ensembles for face detection". In *International Journal of Computer Vision*, 77(1-3):65–86 (2008).

Burt, P. and Adelson, E. "The laplacian pyramid as a compact image code". In IEEE *Transactions on Communications*,, 31(4):532 – 540 (1983).

Cai, D., He, X., Han, J. and Zhang, H.J. "Orthogonal laplacianfaces for face recognition." In IEEE *Transactions on Image Processing*, 15(11):3608–3614 (2006).

Cai, D., He, X. and Han, J. "Spectral regression for efficient regularized subspace learning". In "Proceedings of the IEEE International Conference on Computer Vision", pages 1 –8 (2007).

Cevikalp, H., Neamtu, M. and Wilkes, M. "Discriminative common vector method with kernels". In IEEE *Transactions on Neural Networks*, 17(6):1550–1565 (2006).

Cevikalp, H., Neamtu, M., Wilkes, M. and Barkana, A. "Discriminative common vectors for face recognition". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 27(1):4–13 (2005).

Chen, Y.T. and Chen, C.S. "Fast human detection using a novel boosted cascading structure with meta stages". In IEEE *Transactions on Image Processing*, 17(8):1452 –1464 (2008).

Cootes, T.F., Edwards, G.J. and Taylor, C.J. "Active appearance models". In "Proceedings of the European Conference on Computer Vision", pages 484–498 (1998).

Crosier, M. and Griffin, L. "Texture classification with a dictionary of basic image features". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 1 –7 (2008).

Crowley, J.L. and Berard, F. "Multi-modal tracking of faces for video communications". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 640 –645 (1997).

Crowley, J.L. and Coutaz, J. "Vision for man machine interaction". In "Proceedings of the IFIP TC2/WG2.7 Working Conference on Engineering for Human-Computer Interaction", pages 28–45 (1996). ISBN 0-412-72180-5.

Crowley, J.L. and Parker, A. "A representation for shape based on peaks and ridges in the difference of low-pass transform". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 6(2):156–169 (1984).

Crowley, J.L. and Riff, O. "Fast computation of scale normalised gaussian receptive fields". In "Proc. Scale Space Methods in Computer Vision", pages 584–598 (2003).

Crowley, J.L. and Sanderson, A.C. "Multiple resolution representation and probabilistic matching of 2-d gray-scale shape". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 9(1):113 –121 (1987).

Crowley, J.L. and Stern, R. "Fast computation of the difference of low-pass transform". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 6(2):212–222 (1984).

Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", volume 1, pages 886 –893 vol. 1 (2005).

Fletcher, T. "Relevance vector machines explained". Technical report, University College of London (2010).

Fleuret, F. and Geman, D. "Fast face detection with precise pose estimation". In "International Conference on Pattern Recognition", volume 1, page 10235 (2002).

Freeman, W.T. and Adelson, E.H. "The design and use of steerable filters". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906 (1991).

Freund, Y. and Schapire, R.E. "A decision-theoretic generalization of on-line learning and an application to boosting". In *J. Comput. Syst. Sci.*, 55(1):119–139 (1997).

Fu, Y. and Huang, T.S. "Image classification using correlation tensor analysis". In IEEE *Transactions on Image Processing*, 17(2):226–234 (2008).

Garcia, C. and Tziritas, G. "Face detection using quantized skin color regions merging and wavelet packet analysis". In IEEE *Transactions on Multimedia*, 1(3):264 –277 (1999).

Garcia, C. and Delakis, M. "Convolutional face finder: A neural architecture for fast and robust face detection". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 26:1408–1423 (2004).

Geng, X., Smith-Miles, K., Zhou, Z.H. and Wang, L. "Face image modeling by multilinear subspace analysis with missing values". In IEEE *Transactions on Systems, Man, and Cybernetics*, 41(3):881 –892 (2011).

Geng, X., Zhou, Z.H. and Smith-Miles, K. "Automatic age estimation based on facial aging patterns". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2234–2240 (2007).

Geng, X. and Smith-Miles, K. "Facial age estimation by multilinear subspace analysis". In "IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP", pages 865 –868 (2009).

Georghiades, A.S., Belhumeur, P.N. and Kriegman, D.J. "From few to many: Illumination cone models for face recognition under variable lighting and pose". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660 (2001).

Gourier, N., Hall, D. and Crowley, J.L. "Facial features detection robust to pose, illumination and identity". In "International Conference on Systems Man and Cybernetics", (2004).

Grabner, H., Roth, P.M. and Bischof, H. "Eigenboosting: combining discriminative and generative information". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 18–23 (2007).

Griffin, L. "The second order local-image-structure solid". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1355–1366 (2007).

Griffin, L.D., Lillholm, M., Crosier, M. and van Sande, J. "Basic image features (bifs) arising from approximate symmetry type". In "Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision", pages 343–355 (2009).

Guo, G., Fu, Y., Dyer, C.R. and Huang, T.S. "Image-based human age estimation by manifold learning and locally adjusted robust regression". In IEEE *Transactions on Image Processing*, 17(7):1178–1188 (2008a).

Guo, G., Fu, Y., Dyer, C.R. and Huang, T.S. "A probabilistic fusion approach to human age prediction". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops", (2008b).

Guo, G., Mu, G., Fu, Y. and Huang, T.S. "Human age estimation using bio-inspired features." In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 112–119 (2009).

Hall, D. and Crowley, J.L. "Detection de visages par charactéristiques géneriques calculées à partir des images de luminance". In "Reconnaissance des Formes et Intelligences Artificielle", (2004).

Han, C.C., Liao, H.Y.M., Yu, G.J. and Chen, L.H. "Fast face detection via morphology-based pre-processing". In "Proceedings of the 9th International Conference on Image Analysis and Processing", volume 2, pages 469–476 (1997).

He, X., Yan, S., Hu, Y., Niyogi, P. and Zhang, H.J. "Face recognition using laplacianfaces". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 27:328–340 (2005).

Heiselet, B., Serre, T., Pontil, M. and Poggio, T. "Component-based face detection". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", volume 1, pages I–657 – I–662 vol.1 (2001).

Hjelmas, E. and Low, B.K. "Face detection: A survey". In *Computer Vision and Image Understanding*, 83(3):236 – 274 (2001).

Hua, G., Viola, P.A. and Drucker, S.M. "Face recognition using discriminatively trained orthogonal rank one tensor projections". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (2007).

Huang, C., Ai, H., Li, Y. and Lao, S. "Vector boosting for rotation invariant multi-view face detection". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", volume 1, pages 446 – 453 Vol. 1 (2005a).

Huang, G.B., Ramesh, M., Berg, T. and Learned-Miller, E. "Labeled faces in the wild: A database for studying face recognition in unconstrained environments". Technical Report 07-49, University of Massachusetts, Amherst (2007).

Huang, L.L., Shimizu, A. and Kobatake, H. "Robust face detection using gabor filter features". In *Pattern Recognition*, 26(11):1641 – 1649 (2005b).

Jain, V. and Learned-Miller, E. "Fddb: A benchmark for face detection in unconstrained settings". Technical Report UM-CS-2010-009, University of Massachusetts, Amherst (2010).

Kanade, T. "Picture processing system by computer complex and recognition of human faces". In "doctoral dissertation, Kyoto University", (1973).

Kanade, T. "Computer recognition of human faces". In *Interdisciplinary Systems Research*, 47 (1977).

Keren, D., Osadchy, M. and Gotsman, C. "Anti-faces for detections". In "Proceedings of the European Conference on Computer Vision", pages 134–148 (2000).

Koenderink, J. and van Doom, A. "Representation of local geometry in the visual system". In *Biol. Cybern.*, 55(6):367–375 (1987). ISSN 0340-1200.

Kumanr, R., Banerjee, A. and Vemuri, B. "Volterrafaces: Discriminant analysis using volterra kernels". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (2009).

Kwon, Y. and Vitoria Lobo, N.d. "Age classification from facial images". In *Comput. Vis. Image Underst.*, 74(1):1–21 (1999).

Lampert, C.H. "Kernel methods in computer vision". In *Found. Trends. Comput. Graph. Vis.*, 4:193–285 (2009).

Lanitis., A., Draganova., C. and Christodoulou, C. "Comparing different classifiers for automatic age estimation". In IEEE *Transactions on Systems, Man, and Cybernetics*, 34(1):621–628 (2004).

Lanitis, A., Taylor, C.J. and Cootes, T.F. "Toward automatic simulation of aging effects on face images". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(4):442–455 (2002).

Li, S.Z. and Zhang, Z. "Floatboost learning and statistical face detection". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1112 –1123 (2004).

Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H. and Shum, H. "Statistical learning of multi-view face detection". In "Proceedings of the European Conference on Computer Vision", pages 67–81 (2002).

Lienhart, R., Kuranov, E. and Pisarevsky, V. "Empirical analysis of detection cascades of boosted classifiers for rapid object detection". In "DAGM 25th Pattern Recognition Symposium", pages 297–304 (2003).

Lillholm, M. and Griffin, L. "Novel image feature alphabets for object recognition". In "Proceedings of the International Conference on Pattern Recognition", (2008).

Lindeberg, T. "Scale-space theory in computer vision". In *Kluwer Academic Publishers, Norwell, MA, USA* (1994).

Littlewort, G., Whitehill, J., Wu, T., Fasel, I.R., Frank, M.G., Movellan, J.R. and Bartlett, M.S. "The computer expression recognition toolbox (cert)". In "Proceedings of the IEEE face and gesture recognition", pages 298–305 (2011).

Liu, C. and Shum, H. "Kullback-leibler boosting". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 587–594 (2003).

Liu, C. "Gabor-based kernel pca with fractional power polynomial models for face recognition". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 26(5):572 –581 (2004).

Lowe, D.G. "Distinctive image features from scale-invariant keypoints". In *International Journal of Computer Vision*, 60:91–110 (2004).

Lu, H., Plataniotis, K.N. and Venetsanopoulos, A.N. "Mpca: Multilinear principal component analysis of tensor objects". In IEEE *Transactions on Neural Networks*, 19(1):18–39 (2008).

Lui, Y.M. and Beveridge, J.R. "Grassmann registration manifolds for face recognition". In "Proceedings of the European Conference on Computer Vision", pages 44–57 (2008).

Luo, H. "Optimization design of cascaded classifiers". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 480–485 (2005).

Maydt, J. and Lienhart, R. "Face detection with support vector machines and a very large set of linear features". In "Proceedings of the ACM international conference on Multimedia", (2002).

Mcgurk, H. and Macdonald, J. "Hearing lips and seeing voices". In *Nature*, 264:746–748 (1976).

Meynet, J., Popovici, V. and Thiran, J.P. "Face detection with boosted gaussian features". In *Pattern Recognition*, 40(8):2283–2291 (2007).

Ojala, T., Pietikainen, M. and Harwood, D. "A comparative study of texture measures with classification based on featured distributions". In *Pattern Recognition*, 29:51–59 (1996).

Osadchy, M., Cun, Y.L. and Miller, M.L. "Synergistic face detection and pose estimation with energy-based models". In *Journal of Machine Learning Research.*, 8:1197–1215 (2007).

Osuna, E., Freund, R. and Girosit, F. "Training support vector machines: an application to face detection". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 130 –136 (1997).

O'Toole, A., Cheng, Y., Phillips, P., Ross, B. and Wild, H. "Face recognition algorithms as models of human face processing". In "Proceedings of the IEEE face and gesture recognition", pages 552 –557 (2000).

Pang, Y., Yuan, Y. and Li, X. "Gabor-based region covariance matrices for face recognition". In *IEEE Transactions on Circuits and Systems for Video Technology*, 18(7):989 –993 (2008).

Papageorgiou, C.P., Oren, M. and Poggio, T. "A general framework for object detection". In "Proceedings of the IEEE International Conference on Computer Vision", pages 555–562 (1998).

Pham, D.S. and Venkatesh, S. "Robust learning of discriminative projection for multicategory classification on the stiefel manifold". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (2008).

Phillips, P.J., Moon, H., Rizvi, S.A. and Rauss, P.J. "The feret evaluation methodology for face-recognition algorithms". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104 (2000).

Picard, R.W. *Affective computing*. MIT Press, Cambridge, MA, USA (1997). ISBN 0-262-16170-2.

Ramanathan, N. and Chellappa, R. "Face verification across age progression". In IEEE *Transactions on Image Processing*, 15(11):3349–3361 (2006).

Ramanathan, N., Chellappa, R. and Biswas, S. "Computational methods for modeling facial aging: A survey". In *J. Vis. Lang. Comput.*, 20(3):131–144 (2009).

Ricanek, K.J. and Tesafaye, T. "Morph: A longitudinal image database of normal adult age-progression". In "Proceedings of the IEEE face and gesture recognition", (2006).

Romeny, B.M.t.H., Florack, L., Salden, A.H. and Viergever, M.A. "Higher order differential structure of images". In "Proceedings of the 13th International Conference on Information Processing in Medical Imaging", pages 77–93 (1993).

Rowley, H., Baluja, S. and Kanade, T. "Rotation invariant neural network-based face detection". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (1998).

Roy, A. and Marcel, S. "Haar local binary pattern feature for fast illumination invariant face detection". In "Proceedings of the British Machine Vision Conference", (2009).

Ruan, J. and Yin, J. "Face detection based on facial features and linear support vector machines". In "International Conference on Communication Software and Networks", pages 371–375 (2009).

Sahbi, H. and Boujemaa, N. "Coarse to fine face detection based on skin color adaption". In "Proceedings of the European Conference on Computer Vision", pages 112–120 (2002).

Scherbaum, K., Sunkel, M., Seidel, H.P. and Blanz, V. "Prediction of individual non-linear aging trajectories of faces". In "The European Association for Computer Graphics, 28th Annual Conference, EUROGRAPHICS 2007", volume 26, pages 285–294. Blackwell (2007).

Schiele, B. and Crowley, J. "Recognition without correspondence using multidimensional receptive field histograms". In *International Journal of Computer Vision*, 36:31–50 (2000).

Schiele, B. and Crowley, J.L. "Object recognition using multidimensional receptive field histograms". In "Proceedings of the 4th European Conference on Computer Vision-Volume I - Volume I", Proceedings of the European Conference on Computer Vision, pages 610–619 (1996).

Schiele, B. and Waibel, A. "Gaze tracking based on face-color". In "International Workshop on Automatic Face- and Gesture-Recognition", pages 344–349 (1995).

Schneiderman, H. "Feature-centric evaluation for efficient cascaded object detection". In *Proceedings of the* IEEE *Conference on Computer Vision and Pattern Recognition*, 2:29–36 (2004).

Sinha, P., Balas, B., Ostrovsky, Y. and Russell, R. "Face recognition by humans: Nineteen results all computer vision researchers should know about". In *Proceedings of the IEEE*, 94(11):1948 –1962 (2006).

Sung, K. and Poggio, T. "Example-based learning for view-based human face detection". In "IEEE Transactions on Pattern Analysis and Machine Intelligence", volume 20, pages 39 –51 (1998).

Suo, J., Zhu, S.C., Shan, S. and Chen, X. "A compositional and dynamic model for face aging". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 32(3):385–401 (2010).

Tan, X. and Triggs, B. "Fusing gabor and lbp feature sets for kernel-based face recognition". In "Proceedings of the 3rd international conference on Analysis and modeling of faces and gestures", AMFG'07, pages 235–249 (2007).

Tao, D., Li, X., Wu, X. and Maybank, S. "General tensor discriminant analysis and gabor features for gait recognition". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1700 –1715 (2007).

Tipping, M.E. "Sparse bayesian learning and the relevance vector machine". In *Journal of Machine Learning Research*, 1:211–244 (2001).

Tola, E., Lepetit, V. and Fua, P. "Daisy: An efficient dense descriptor applied to wide baseline stereo". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 99(1) (2009). ISSN 0162-8828.

Turk, M. and Pentland, A. "Face recognition using eigenfaces". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 586–591 (1991a).

Turk, M. and Pentland, A. "Eigenfaces for recognition". In *J. Cognitive Neuroscience*, 3(1):71–86 (1991b).

Tuzel, O., Porikli, F. and Meer, P. "Region covariance: A fast descriptor for detection and classification". In "Proceedings of the European Conference on Computer Vision", pages 589–600 (2006).

Valentin, D., Abdi, H. and Edelman, B. "What represents a face: A computational approach for the integration of physiological and psychological data". In *Perception*, 26(10):1271 – 1288 (1997).

Viola, P. and Jones, M. "Rapid object detection using a boosted cascade of simple features". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", volume 1, pages I–511 – I–518 vol.1 (2001).

Viola, P. and Jones, M.J. "Robust real-time face detection". In *International Journal of Computer Vision*, 57(2):137–154 (2004).

Waring, C. and Liu, X. "Face detection using spectral histograms and svms". In IEEE *Transactions on Systems, Man, and Cybernetics*, 35(3):467 –476 (2005).

Winder, S.A.J. and Brown, M. "Learning local image descriptors". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (2007).

Witkin, A. "Scale-space filtering: A new approach to multi-scale description". In "IEEE International Conference on Acoustics, Speech, and Signal Processing.", volume 9, pages 150 – 153 (1984).

Wolf, L., Hassner, T. and Taigman, Y. "Similarity scores based on background samples". In "ACCV (2)", pages 88–97 (2009).

Wright, J. and Hua, G. "Implicit elastic matching with random projections for pose-variant face recognition". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (2009).

Wu, J. and Rehg, J. "Where am i: Place instance and category recognition using spatial pact". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 1–8 (2008).

Wu, J., Brubaker, S., Mullin, M. and Rehg, J. "Fast asymmetric learning for cascade face detection". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30(3):369 –382 (2008).

Xiao, R., Li, M.J. and Zhang, H.J. "Robust multipose face detection in images". In IEEE *Transactions on Circuits and Systems for Video Technology*, 14(1):31 – 41 (2004).

Xiaohua, L., Lam, K.M., Lansun, S. and Jiliu, Z. "Face detection using simplified gabor features and hierarchical regions in a cascade of classifiers". In *Pattern Recognition*, 30(8):717–728 (2009).

Xie, X., Zheng, W.S., Lai, J. and Yuen, P. "Face illumination normalization on large and small scale features". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 1–8 (2008).

Yan, S., Wang, H., Tang, X. and Huang, T.S. "Learning auto-structured regressor from uncertain nonnegative labels". In "Proceedings of the IEEE International Conference on Computer Vision", (2007a).

Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X. and Zhang, H. "Multilinear discriminant analysis for face recognition". In IEEE *Transactions on Image Processing*, 16(1):212 –220 (2007b).

Yan, S., Shan, S., Chen, X. and Gao, W. "Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pages 1 –7 (2008a).

Yan, S., Wang, H., Tang, X., Liu, J. and Huang, T.S. "Regression from uncertain labels and its applications to soft biometrics". In IEEE *Transactions on Information Forensics and Security*, 3(4):698–708 (2008b).

Yang, G. and Huang, T. "Human face detection in a complex background". In *Pattern Recognition*, 27(1):53 – 63 (1994).

Yang, J., Zhang, D., Frangi, A.F. and yu Yang, J. "Two-dimensional pca: A new approach to appearance-based face representation and recognition". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137 (2004).

Yang, M.H., Kriegman, D.J. and Ahuja, N. "Detecting faces in images: A survey". In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58 (2002).

Yokono, J. and Poggio, T. "Oriented filters for object recognition: an empirical study". In "Proceedings of the IEEE face and gesture recognition", (2004).

Young, R.A., Lesperance, R.M. and Meyer, W.W. "The gaussian derivative model for spatial-temporal vision: I. cortical model." In *Spatial Vision*, 2001:3–4 (2001).

Zhang, B., Shan, S., Chen, X. and Gao, W. "Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition". In IEEE *Transactions on Image Processing*, 16(1):57–68 (2007).

Zhang, C. and Zhang, Z. "A survey of recent advances in face detection". Technical Report MSR-TR-2010-66, Microsoft Research (2010). Available at http://research.microsoft.com/apps/pubs/default.aspx?id=132077.

Zhang, W., Shan, S., Gao, W., Chen, X. and Zhang, H. "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition". In "Proceedings of the IEEE International Conference on Computer Vision", (2005).

Zhao, W., Chellappa, R., Rosenfeld, A. and Phillips, P.J. "Face recognition: A literature survey". In *ACM Computing Surveys*, pages 399–458 (2003).

List of Figures

2.1 Automatic face verification system 14

2.2 Hierarchy of resolution images proposed by (Yang and Huang, 1994). Each square cell consists of $n \times n$ pixels in which intensity of each pixel is replaced by the average intensity of the pixels in that cell. (Courtesy of Yang *et al.* (2002)) 15

2.3 (a) Wavelet decomposition performed over a facial candidate region. (b) Final regions after take into account restrictions and probabilistic metrics (Courtesy of Garcia and Tziritas (1999)) 16

2.4 Diagram of skin region selection proposed by Sahbi and Boujemaa (2002) (Courtesy of Sahbi and Boujemaa (2002)) 17

2.5 Example of facial template and Overall face detection system proposed by Heiselet *et al.* (2001) (Courtesy of Heiselet *et al.* (2001)) . . . 17

2.6 Overall face detection system proposed by Hall and Crowley (2004). (Courtesy of Hall and Crowley (2004)) 18

2.7 Energy model and neural network architecture proposed by Osadchy et al Osadchy *et al.* (2007) (Courtesy of Osadchy *et al.* (2007)) . . . 20

2.8 Haar-like feature sets.(a)Original Feature set proposed by Viola and Jones (2001) and oriented Haar-like features proposed by Lienhart *et al.* (2003). (b)Sparse features represented in granular space proposed by Li *et al.* (2002) 22

2.9 (a)Anisotropic Gaussian Filters proposed by Meynet *et al.* (2007) (Courtesy of Meynet *et al.* (2007)). (b)Facial images based in Gabor filter as proposed by Huang *et al.* (2005b)(Courtesy of Huang *et al.* (2005b)) 22

2.10 Viewpoint face recognition by patches proposed by Ashraf *et al.* (2008). (Courtesy of Ashraf *et al.* (2008)). The patches from the probe image are compared with the gallery set using a probabilistic model 24

2.11 Example of eigenfaces proposed by Turk and Pentland (1991a). (a) Original dataset (b) First twenty eigenfaces from the original dataset. 25

2.12 Local Binary Pattern proposed for recognizing faces by Ahonen *et al.* (2006). (a) Original dataset (b) Result images after applying LBP operator in the original dataset. 25

2.13 Training and patch images used by Kumanr *et al.* (2009) to project the original feature space in other more discriminative space using volterra kernels(Courtesy of Kumanr *et al.* (2009)) 27

2.14 AGing pattErn Subspace (AGES) proposed by Geng *et al.* (2007). (a) Feature extraction from a time-ordered sequence of facial images. (b) Reconstruction of empty sub-space position to estimate age. (Courtesy of Geng *et al.* (2007)) 28

2.15 Age simulation example using the model proposed by Scherbaum *et al.* (2007). A 3-D face model of a boy is transformed into a face at an appropriate target age, then the image is rendered in an appropriate image-background. (Courtesy of Scherbaum *et al.* (2007)) 28

2.16 Age simulation example using the model proposed by Suo *et al.* (2010). An input image is progressively transformed in a set of unseen facial ages. (Courtesy of Suo *et al.* (2010)) 30

2.17 Unified facial analysis architecture proposed in this thesis 32

3.1 Gaussian scale space representation at different values of σ. Image details are eliminated when the scale value increases. (image from `http://th.physik.uni-frankfurt.de/~jr/physlist.html`) . 37

3.2 Impulse responses for steering basis of Gaussian derivatives computed at $\sigma = \sqrt{2}$ 38

3.3 Normalized power spectra for Gaussian derivatives up to fourth order computed at $\sigma = 1, 2, 4$ and 8. Gaussian derivatives act like band-pass filters 40

4.1 Overview of the cascade architecture. At each stage, the classifier either rejects the sample and the process stops, or accepts it and the sample is forwarded to the next stage 50

4.2 Example images from the sensitive test dataset (images modified from LFW (Wolf *et al.*, 2009; Huang *et al.*, 2007)) 55

4.3 Example images from the MIT + CMU face dataset 55

4.4 Example images from the FDDB face dataset 55

4.5 Node performances for the four cascades trained with the feature sets show in the table 4.1. The node error decreases when the derivative order rises, specially for deeper nodes in the cascade. Notice also that the cascade trained using the feature set 4 has only 21 nodes which show a high performance of the features to reach the fixed node detection and positive rates 58

4.6 Graphical example of our speed-optimized cascade framework for different values of p. The distribution of features in each node of the cascade takes in consideration the computational cost of each derivative order 60

4.7 Accumulative error rate across the cascade nodes using $p = \{1, 2, 3, 4\}$ in the speed-optimized cascade training framework. Adding higher derivative order decreases the error rate, specially after the node number 5. 60

4.8 Results of comparing a non-optimized cascade of Gaussian derivatives with a cascade of Haar features in the sensitivity testing dataset. In this experiment Gaussian derivatives showed a high invariance to rotation angle (a) and blurring (b) in the other hand a high invariance to image noise is exhibit by Haar features 62

4.9 Performance comparison with previous works in the CMU+MIT face dataset 62

4.10 Performance comparison of a non-optimized cascades of Gaussian derivatives features with a Haar-features cascade in the FDDB face dataset. As we can see, the non-optimized cascade of Gaussian derivatives outperform in this dataset. 64

4.11 Computational Load comparison between a cascade of Haar features and a non-optimized cascade of Gaussian derivatives. The number of accumulative requests (b) for a cascade with Gaussian derivatives is inferior to a cascade with Haar features, as consequence the estimated load (c) in a cascade with Gaussian is reduced, specially in the first nodes. 64

4.12 Computational load shown as function of image location: a) Image Original, b)with a cascade of Haar features and c) with a non-optimized cascade of Gaussian derivatives. notice than the sub-window positions with a level of intensity higher have a high computational load. In the case of Gaussian derivatives features the computational load is reduced as we can see by its intensity levels (c) 65

4.13 Results of comparing different optimized cascades of Gaussian derivatives in the sensitivity testing dataset. No considerable differences in performance between the non-optimized and the optimized model were noticed. 67

4.14 Performance comparison of optimized-speed cascades of Gaussian derivatives features in the CMU+MIT face dataset 68
4.15 Performance comparison of speed-optimized cascades of Gaussian derivatives features in the FDDB face dataset 68
4.16 Computational Load comparisons in the optimized-cascade framework for different values of p 69
4.17 Computational load shown as function of image location for different values of p in the speed-cascade framework 69
4.18 Example of detections using a non-optimized cascade of Gaussian Derivatives. (a) MIT+CMU and (b) FDDB face dataset 70

5.1 A set of Gaussian filters is computed from the input pyramid then LBP is applied to obtain Binary Gaussian Jet Maps 78
5.2 Each local binary map is divided into non-overlapping rectangular sub-regions with a specific size. A set of histograms is then computed for each sub-region 78
5.3 Example of construction for one tensor in our method. Each histogram is arranged in a third order tensor 78
5.4 Fusing tensors with MPCA. (a) MPCA is applied at each tensor and then the resulting vectors are fused into one. (b) The tensors are fused into one before applying MPCA 81

6.1 Examples of images from the FERET data set 87
6.2 Example of images from the Yale face database 88
6.3 Example of images from the Yale B + Extended Yale B dataset 89
6.4 Face recognition architectures proposed in this thesis using tensors of HBGM. (a)concatenated vectors then KDCV+NN. (b) concatenated tensors then MPCA and finally KDCV+NN 90
6.5 Binary Gaussian maps at different scales (each column) of two images with different conditions of illumination ((a) and (b)), the three last rows correspond to $Mag_\theta\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}\right)$, $LoG_\theta\left(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}\right)$ and $\gamma_\theta(x, y, \sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4, \})$ respectively 94
6.6 Cumulative Match Curves for the vectors y_1, y_2 and y_3 using Mag, LoG and γ on the four FERET probe sets (a)f_b (b) f_c, (c) Dup_I and (d)Dup_{II} 95
6.7 Rank-1 recognition accuracy versus dimensionality reduction with MPCA for the vectors y_1 , y_2 and y_3 using Mag, LoG and γ on the four FERET probe sets (a) fb , (b)fc, (c)dup I and (d)dup II. 96
6.8 Cumulative Match Curves for the vectors y_T and y_F using Mag, LoG and γ on the four FERET probe sets (a)f_b (b) f_c, (c) Dup_I and (d)Dup_{II} 97

6.9 Rank-1 recognition accuracy versus dimensionality reduction with MPCA for the vectors y_T and y_F using Mag, LoG and γ on the four FERET probe sets (a)f_b (b) f_c, (c) Dup_I and (d)Dup_{II} 98

7.1 Age estimation architectures proposed in this thesis using tensors of HBGM. (a)concatenated vectors(y_T) then RVM. (b) concatenated tensors then MPCA (y_F)and finally RVM 107

7.2 Sample images of different individuals from the FG-NET dataset . . 107

7.3 Sample images of different individuals from the MORPH dataset . . 108

7.4 Cumulative scores on (a)FG-NET database and (b)MORPH (test) database 112

A.1 The $O(N)$ cascade convolution pyramid algorithm 121

A.2 The $\sqrt{2}$ resampling Operator, $S_{\sqrt{2}}\{\cdot\}$, selects even columns of even rows and odd columns of odd rows. 121

A.3 The $\sqrt{2}$ expansion operator, $E\sqrt{2}\{\cdot\}$, maps rows of a filter onto diagonals, increasing sample distance by $\sqrt{2}$, illustrated here for the 3×3 filter $B_2(i, j)$. 122

A.4 Illustration of diagonal sampling with levels k=1, 2, 3, 4 from a diagonally sampled pyramid. The symbol "+" represents image samples. Each image has half the pixels of the previous image. 123

A.5 The pyramid buffer is composed of $P = N(1 + 1/2 + 1/4 + 1/8 + \cdots) = 2N$ samples. 124

A.6 Impulse responses of Gaussian derivatives up to the fourth order computed with a Half-Octave Gaussian Pyramid 124

List of Tables

3.1 Average (10.000 runs) pyramid construction time for five different input data sizes (in pixels), on an Intel®Pentium ®, 2.80 Ghz 42

4.1 Four different feature sets using different Gaussian derivative orders at pyramid levels of $\sigma = \left\{\sqrt{2}, 2, 2\sqrt{2}\right\}$ and orientations $\theta = \{0, \pi/4, \pi/2, 3\pi/4\}$ 57

4.2 A comparison of detection rates on the CMU+MIT data set for several standard detectors 63

6.1 The Rank-1 Recognition Rates of Different Algorithms on the FERET Probe Sets 91

6.2 The Rank-1 Recognition Rates of Diferent Algorithms on The YALE Face Dataset 92

6.3 The Rank-1 Recognition Rates of Different Algorithms on The YALE B+EXTENDED YALE-B Face Dataset 93

6.4 CPU average times of each step in our face recognition method using Mag, LoG and γ 94

6.5 The Rank-1 Recognition Rates of Different Algorithms on the FERET Probe Sets 97

6.6 The Rank-1 average recognition error rates on the Yale B + Extended Yale B dataset with different training set sizes 100

7.1 MAEs on the FG-NET database for different tensorial configurations 109

7.2 MAE (years) at different age groups on FG-NET. 109

7.3 MAE (years) comparisons on FG-NET 110

7.4 MAE (years) comparisons on MORPH (Test Set) 110

List of Algorithms

1 The cascade framework (Wu *et al.*, 2008) 50
2 Detecting faces with a cascade of classifiers using a sliding window . 52
3 The speed-optimized cascade framework 59
4 MPCA algorithm proposed by Lu *et al.* (2008) 76
5 Kernel Discriminative Common Vectors (Cevikalp *et al.*, 2006). . . . 87
6 Relevance Vector Machines (Tipping, 2001). (courtesy of Fletcher (2010)) 105