THÈSE

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
UFR des sciences fondamentales et appliquées
XLIM-SIC
(Diplôme National - Arrêté du 7 août 2006)

École doctorale : Sciences et ingénierie pour l'information, mathématiques - S2IM (Poitiers)
Secteur de recherche : Traitemement du signal et des images

Présentée par :
Cristina Bordei

Face analysis using polynomials

Directeur(s) de Thèse :
Philippe Carré, Pascal Bourdon, Bertrand Augereau

Soutenue le 03 mars 2016 devant le jury

Jury :

Président	**Kidiyo Kpalma**	Professeur des Universités, INSA de Rennes
Rapporteur	**Kidiyo Kpalma**	Professeur des Universités, INSA de Rennes
Rapporteur	**Renaud Séguier**	Professeur des Universités, Supélec de Rennes
Membre	**Philippe Carré**	Professeur des Universités, Université de Poitiers
Membre	**Pascal Bourdon**	Maître de conférences, Université de Poitiers
Membre	**Bertrand Augereau**	Maître de conférences, Université de Poitiers
Membre	**Pierre Chainais**	Maître de conférences, École centrale de Lille

Pour citer cette thèse :
Cristina Bordei. *Face analysis using polynomials* [En ligne]. Thèse Traitemement du signal et des images. Poitiers : Université de Poitiers, 2016. Disponible sur Internet <http://theses.univ-poitiers.fr>

THÈSE

pour l'obtention du Grade de
DOCTEUR DE L'UNIVERSITE DE POITIERS
(Faculté des Sciences Fondamentales et Appliquées)
(Diplôme National - Arrêté du 7 août 2006)

École Doctorale: Sciences et Ingénierie pour l'Information, Mathématiques(S2IM)

Secteur de recherche : Traitement du Signal et des images

Présentée par:

Cristina BORDEI

****

FACE ANALYSIS USING POLYNOMIALS

****

Directeur de thèse: Philippe CARRÉ

Co-Encadrants de thèse: Bertrand AUGEREAU, Pascal BOURDON

****

Soutenue le 3 mars 2016
devant la Commission d'Examen composée de:

****

MEMBRES DU JURY

M.	Renaud SÉGUIER, Professeur des Universités, Supélec Rennes,	Rapporteur
M.	Kidiyo KPALMA, Professeur des Universités, INSA de Rennes,	Rapporteur
M.	Pierre CHAINAIS, Maître de Conférences HDR, Ecole Centrale de Lille,	Examinateur
M.	Philippe CARRÉ, Professeur des Universités, Université de Poitiers	Examinateur (directeur de thèse)
M.	Bertrand AUGEREAU, Maître de Conférences, Université de Poitiers,	Examinateur (co-encadrant)
M.	Pascal BOURDON, Maître de Conférences, Université de Poitiers,	Examinateur (co-encadrant)

Résumé

Considéré comme l'un des sujets de recherche les plus actifs et visibles de la vision par ordinateur, de la reconnaissance des formes et de la biométrie, l'analyse faciale a fait l'objet d'études approfondies au cours des deux dernières décennies. Toutefois, en pratique il reste un problème difficile en raison de variations de pose, d'éclairage , d' occlusions ou d'un environnement non-contrôlé etc. Diverses approches ont été proposées pour l'extraction et la modélisation de caractéristiques du visage en termes de robustesse, de coût de calcul et de la précision, chacune comportant des avantages et des inconvénients. Le travail de cette thèse a pour objectif de proposer de nouvelles techniques d'utilisation de représentations de texture basées polynômes pour l'analyse faciale.

La première partie de cette thèse, est dédiée à l'intégration de bases de polynômes dans les modèles actifs d'apparence - un ensemble d'outils statistiques utilisés pour modéliser la forme et l'apparence d'un objet qui ont prouvé leur efficacité pour la modélisation faciale. Nous proposons dans un premier temps une manière d'utiliser les coefficients obtenus après projections polynomiale dans la modélisation de l'apparence. Deux approches différentes pour remplacer la représentation de texture originale sont détaillées – calculés soit sur des régions d'intérêts situées autour de points annotés, soit à partir d'une décomposition polynomiale multi-résolution de la texture alignée. Ensuite, afin de réduire la complexité du modèle et puisque la représentation polynomiale d'une image est multi-échelle, nous proposons de choisir et d'utiliser les meilleurs coefficients polynomiaux en tant que représentation de texture. En utilisant un algorithme de régression itératif s'appuyant sur des coefficients polynomiaux compressées nous avons obtenu de très bons résultats d'alignement de visage démontrant la compacité de notre représentation. Enfin, nous montrons comment, outre l'utilisation des coefficients polynomiaux pour la modélisation de texture ils peuvent être utilisés dans un algorithme de descente de gradient étant donné que la décomposition polynomiale est équivalente à un banc de filtres.

La deuxième partie de la thèse porte sur l'utilisation des bases polynomiales pour la détection des points/zones d'intérêt et comme descripteur pour la reconnaissance des expressions faciales. Inspirés par des techniques de détection des singularités dans des champ de vecteurs, nous commençons par présenter un algorithme utilisé pour l'extraction des points d'intérêt dans une image. Notre approche consiste en deux grandes étapes - la détermination du champ de normales de l'image suivi par la recherche de points d'intérêt dans ce champ, toutes deux présentées dans le contexte général d'un schéma multi-échelle et multi-résolution. Enfin, nous montrons comment les bases polynomiales peuvent être utilisées pour extraire des informations sur les expressions faciales. Puisque les coefficients polynomiaux fournissent une analyse précise multi-échelles et multi-orientation et traitent le problème de redondance

efficacement ils sont utilisés en tant que descripteurs dans un algorithme de classification d'expression faciale. Les résultats expérimentaux confirment que notre approche fonctionne bien dans ce contexte, tout en étant performante et donnant des résultats de haute précision.

Abstract

As one of the most active and visible research topic in computer vision, pattern recognition and biometrics, facial analysis has been extensively studied in the past two decades. Yet it is still a challenging problem in practice due to uncontrolled environment, occlusions and variations in pose, illumination, etc. Various methods have been proposed for facial features extraction, with different advantages and drawbacks in terms of robustness, computational cost and accuracy. The work in this thesis presents novel techniques to use polynomial basis texture representations for facial analysis.

The first part of this thesis, is dedicated to the integration of polynomial bases in the Active Appearance Models - a set of statistical tools used to model the shape and appearance of an object that proved to be very efficient in modeling faces. First we propose a way to use the coefficients obtained after polynomial projections in the appearance modeling. Two different schemes to replace the original texture representation are detailed - calculated on texture patches sampled around key landmarks, or retrieved from a multi-resolution polynomial decomposition of the full aligned texture. Then, in order to reduce model complexity and since the polynomial representation of an image is multi-scale we proposed to select and use as a texture representation the strongest polynomial coefficients. Using a cascaded regression algorithm based on compressed polynomial coefficients we obtained very good alignment results demonstrating the compactness of our representation. Finally we show how in addition to the texture representation polynomial coefficients can be used in a gradient descent algorithm since polynomial decomposition is equivalent to a filter bank.

The second part of the thesis concerns the use of the polynomial bases for interesting points and areas detection and as a descriptor for facial expression recognition. We start by presenting an algorithm used for accurate image keypoints localization inspired by techniques of singularities detection in a vector field. Our approach consists in two major steps -the calculation of an image vector field of normals and the keypoint selection within the field both presented in a multi-scale multi resolution scheme. Finally we show how polynomial bases can be used to extract informations about facial expressions. Since polynomial coefficients provide precise multi-scale and multi-orientation analysis and handle the redundancy problem effectively they are used as descriptors in an facial expression classification algorithm. Experimental results confirm that our approach performs well in this context, being computationally efficient and giving high accuracy results.

Résumé détaillé

Parce qu'elles fournissent un formalisme théorique efficace pour l'analyse multi-échelles et multi-orientations, les ondelettes sont efficaces pour traiter les problèmes de changements d'éclairage et de pose, et sont largement utilisées dans des applications d'analyse faciale. Encouragés par les résultats de l'utilisation des bases polynomiales pour la modélisation des champs de vecteurs et l'analyse de mouvements simples du visage nous proposons d'étudier et d'utiliser une représentation similaire à la représentation en ondelettes, mais plus souple et adaptative : la transformée polynômiale.

Analyse d'image par bases complètes

Nous avons tout d'abord commencé par une étude de la représentation polynomiale 2D d'une image et montré comment les coefficients obtenus à partir de projections des intensités lumineuses d'une image sur une base polynomiale complète peuvent être utilisés pour une approximation hiérarchique et compacte du signal image, et pour son analyse structurelle.

La technique présentée pour la décomposition polynomiale multi-resolution d'une image offre une réelle souplesse, notamment vis-à-vis du choix des facteurs de résolution, qui peuvent être indépendants entre niveaux de décomposition. Par conséquent, la transformée polynomiale multi-échelle est plus compacte qu'une représentation en ondelettes (de type Gabor, typiquement utilisée dans l'analyse faciale), permettant de faire disparaître la plupart des problèmes d'échantillonnage, comme le compromis entre l'échantillonnage fréquentiel et d'orientations.

De plus, les bases complètes permettent d'obtenir, pour un ensemble donné de valeurs, une fonction interpolatrice qui est un polynôme d'osculation du premier ordre (*ie* tel que $\forall \mathbf{x}_{(u,v)} \in D, P_I\left(\mathbf{x}_{(u,v)}\right) = I\left(\mathbf{x}_{(u,v)}\right)$). Par ailleurs, on peut considérer la projection sur $B_{i,j}$ comme un opérateur de différences finies multi-échelle relatif à la différentiation $\partial_1^i \partial_2^j$.

Deux techniques de sélection de coefficients polynomiaux pour l'approximation d'une image ont été proposées et démontrent l'efficacité de notre approche en la comparant aux ondelettes de Haar, CDF 9/7 et à la décomposition en valeurs singulières.

Modèles actifs d'apparence polynomiaux

Notre motivation pour utiliser une représentation polynomiale dans les AAM (modèles actifs d'apparence) vient du fait que les polynômes orthogonaux présentent certaines propriétés liées au système visuel humain [Bla74], notamment une représentation multi-échelle /multi-résolution de l'information. De plus, une image pourrait être approximée à partir des

coefficients polynomiaux en ne conservant qu'un nombre défini de coefficients assurant une certaine énergie cumulée, similaire à l'analyse en composantes principales.

Nous avons commencé cette partie par une présentation détaillée des modèles actifs d'apparence, un modèle statistique permettant de faire conjointement l'analyse et la synthèse d'une classe d'objets à partir d'un ensemble d'apprentissage comprenant différentes vues d'un objet. L'algorithme AAM comprends 2 étapes principales - la modélisation des données et l'ajustement du modèle, et nous allons voir par la suite comment il est possible d'intégrer les bases polynomiales dans chacune de ces étapes.

Une texture polynomiale pour les modèles actifs d'apparence

Tout d'abord une nouvelle approche pour la représentation de la texture dans les modèles actifs d'apparence est présentée. Celle-ci est basée sur l'utilisation de coefficients issus de projections des intensités lumineuses sur une base polynomiale complète.

Afin d'améliorer la robustesse du processus d'ajustement des AAM, l'idée est de remplacer le mode de représentation de texture du modèle de référence par des projections polynomiales sur une base complète orthonormée. Ceci revient à calculer un modèle d'apparence en remplaçant le vecteur des intensités pixels en entrée de l'ACP (analyse en composantes principales) par un vecteur de coefficients obtenus par projections polynomiales dans la base complète sur des textures alignées.

Deux possibilités se présentent pour le calcul du vecteur de coefficients : il pourra être effectué soit sur des régions d'intérêts situées autour de points annotés (PAAM), soit à partir d'une décomposition polynomiale multi-résolution de la texture, suivie d'une étape éventuelle de quantification (FT-PAAM). Dans l'approche PAAM, le modèle AAM donne des résultats plus précis que celui calculé sur l'ensemble des pixels du modèle AAM, en particulier dans le cas de la variation d'expression faciale ou pose. Comme nous utilisons des modèles de texture spatialement localisés autour des points d'intérêt, notre méthode offre obligatoirement plus de robustesse aux modification locales de texture. Pour l'approche FT-PAAM, étant donné que l'on obtient une représentation hiérarchique de l'information lors de la transformation des coefficients de texture via des projections polynômiales nous observons donc que les points d'intérêt sont déterminés avec une meilleure précision par rapport aux autres modèles, en particulier pour les points sur le menton, qui sont assez difficiles à situer et qui ne sont généralement pas pris en compte dans les calculs d'erreurs.

Nous avons ensuite développé le travail de Wolstenholme et Stegmann [WT99] appliqué à l'alignement de visage où des sous-ensembles de coefficients d'ondelettes ont été mod-

élisés au lieu des intensités des pixels. En déviant de leur approche, nous avons inclus les coefficients d'approximation polynomiaux dans le cadre de la régression. La méthode proposée repose sur deux éléments principaux: l'adaptation des AAM afin d'incorporer des caractéristiques de texture et donc de la génération de ces dernières en utilisant les bases polynomiales et de l'algorithme de régression utilisé pour l'ajustement du modèle.

L'algorithme CDAAM est donc spécifié, permettant l'intégration de la compression polynômiale dans un cadre de régression, en utilisant les paramètres globaux de la forme conjointement avec les paramètres combinés de formes et de l'apparence. Les coefficients d'approximation polynomiale sont utilisés pour compresser les données, puis une analyse en composantes principales est utilisée pour réduire la dimensionnalité de données, similaire aux AAM traditionnels.

Des expériences en utilisant différentes bases polynomiales pour sept rapports de compression différents ont été effectuées. Il a été constaté que notre méthode permet d'obtenir une précision d'alignement très stable et de très bons résultats d'alignement tout en augmentant le taux de compression et en maintenant un faible pourcentage de données. En comparant notre approche avec les ondelettes de Haar et aux ondelettes CDF 9/7 nous avons conclu que pour des taux de compression élevés la méthode utilisant les coefficients polynomiaux offre les meilleurs résultats.

Les expériences d'alignement de modèle sur des images de quatre bases d'images confirment que les deux modèles de texture proposés, ainsi que les modèles compressés permettent d'obtenir une meilleure précision d'alignement. Nos résultats sont très satisfaisants et montrent que par ses propriétés - sa paramétrisation simple et sa souplesse, la représentation polynomiale est un substitut prometteur aux représentations classiques de texture.

Algorithme de descente de gradient en utilisant les bases polynomiales

Dans le chapitre précédent, nous avons proposé une amélioration de l'aspect texture dans le cadre AAM. Toutefois, nous avons vu que l'approche de décomposition polynomiale multi-résolution est équivalente à un banc de filtres, donc les coefficients polynomiaux peuvent être utilisés dans un algorithme de descente de gradient.

Nous avons tout d'abord vérifié la validité de notre idée en utilisant les coefficients obtenus par des projections polynomiales dans la méthode compositionelle inverse de Matthews et Baker [MB04]. Les resultats préliminaires ont montré l'efficacité de notre approche.

Nous avons ensuite reformulé l'algorithme compositionnel inverse afin d'avoir un ajustement à travers de multiples réponses de filtre polynomiaux. En modifiant la fonction d'erreur nous avons montré que l'intégration de la représentation polynomiale peut être directement inclue dans la cadre de l'algorithme de Lucas Kanade.

Tout d'abord nous avons intégré la transformée polynomiale dans l'algorithme utilisant une approximation de Taylor d'ordre 1 à savoir l'algorithme Gauss Newton. En utilisant les bases polynomiales au lieu de minimiser la somme des différences des carrés entre une image constante (le modèle) et l'image exemple par rapport aux paramètres de transformation, la différence entre l'image et le modèle correspondant calculé par des projections dans la base polynomiale complète est minimisée.

Matthews et Baker ne recommandent pas d'utiliser la méthode de Newton (celle utilisant une approximation de Taylor d'ordre 2) parce que cette approche utilise une estimation sophistiquée de la matrice Hesienne qui de plus est présumée sans bruit. Ils affirment également que l'augmentation du bruit dans l'estimation des dérivées secondes du modèle l'emporte sur la sophistication accrue dans l'algorithme. Les projections dans la base de polynômes comprennent une convolution avec des fonctions de pondération utilisés pour la construction de la base polynomiale. En utilisant une base d'Hermite, l'image d'entrée est convoluée avec un filtre Gaussien qui limite le bruit dans le calcul des gradient et des dérivées secondes. Par conséquent, nous présentons l'approche de Newton en utilisant des projections sur les premier et second ordre d'une base de polynômes.

Deux extensions de l'algorithme d'alignement d'image de composition inverse utilisant des projections de polynômes on étés presentées ci-dessus. À notre connaissance, nous proposons la première solution unifiée qui traite la descente de gradient ainsi que la représentation de la texture dans un seul modèle cohérent.

Les algorithmes ont été évalués sur des ensembles de données complexes, y compris la base de donnés Multi-PIE qui combine une variabilité en identité, pose, expression faciale et la variation d'éclairage. En utilisant les projections polynomiales sur la majorité des bases d'images on obtient de résultats d'alignement améliorés.

De plus en utilisant notre approche, les erreurs moyennes de la méthode de Newton sont nettement inférieures à celles de la méthode de Gauss Newton.

Nous croyons que le cadre présenté est une base solide pour explorer des modèles plus complexes de visage, et qui pourrait permettre d'améliorer davantage la qualité d'alignement dans les images / vidéos.

Des points d'intérêt aux expressions faciales

Après avoir travaillé sur le suivi de points d'intérêt dans un visage nous allons nous intéresser dans cette partie à la détection des points/zones d'intérêt dans une image ainsi qu'à l'utilisation des coefficients obtenus par projection polynomiale en tant que descripteur pour la reconnaissance des expressions faciales.

Détection des points d'intérêt dans la caractérisation des textures faciales

Étant donné que les bases polynomiales ont été utilisées pour la détection et la caractérisation des singularités dans un champ de vecteurs, nous proposons de les utiliser pour la localisation précise des points d'intérêt dans les images couleurs, points déduits des singularités du champ des normales.

Nous présentons donc un algorithme dans lequel le processus de détection des points d'intérêt dans une image repose sur deux phases fondamentales, la détermination du champ des normales et la recherche des singularités dans ce champ. Chaque phase est détaillée dans le contexte général d'un schéma multi-échelle et multi-résolution.

L'approche présentée fonctionne sur des images couleur et en niveaux de gris, le nombre de points d'intérêt détectés est ajustable sur une large plage par des seuils très simples et donne la possibilité d'utiliser différents types de bases (et donc d'utiliser différents types de lissage dans la construction des pyramides d'échelles) pendant la création de la base multivariée.

La qualité de notre détecteur est ensuite évaluée sur la base d'images Oxford [MS05] et quelques séquences de la base de Jared Heinly [HDF12] sur des transformées d'images contrôlées (en rajoutant artificiellement du bruit, rotation,des transformées d'illumination ou d'échelle) ainsi que sur de la mise en correspondance sur des images réelles, en utilisant tout l'ensemble de données conjointement aux matrices d'homographie.

Suite aux expériences nous avons conclu que notre approche est robuste aux changements d'échelle, d'éclairage et de flou. Cependant, au vu des résultats, même si une orientation est calculée pour chaque point-clé, notre détecteur de points d'intérêt est sensible aux rotations, des modifications devraient donc être apportées à notre méthode d'affectation d'orientation.

Pendant l'étape d'évaluation de notre détecteur, nous avons remarqué que par rapport aux détecteurs SIFT et SURF sur les images de visage les zones sélectionnées avaient une signification sémantique, notre détecteur ayant toujours choisi les yeux, le nez et la bouche et ignorant les zones avec une texture constante telles que les joues ou de la peau du front. Vu que pendant le processus de recherche, nous ne gardons que les points dominants et robustes et que dans les images de visages ces points correspondent à des régions clés, nous avons implémenté un algorithme d'alignement qui utilise les zones d'intérêt détectées par notre approche.

L'algorithme mis en oeuvre utilise une régression en cascade similaire à celui utilisé dans le modèle AAM compressé avec les zones d'intérêt en tant que fonctionnalités pour l'algorithme de régression.

Les résultats montrent que pour les bases de données comportant des rotations faciales notre approche n'est pas pertinente (suite au problème d'orientation des points détectés) et que pour des bases de données où le visage est de taille raisonnable (supérieure à 300 pixels tel dans la base de données MUG) notre approche donne de meilleurs résultats que celle utilisant toute la texture du visage.

Représentation de texture polynomiale pour la reconnaissance de l'expression faciale

Alors que de nombreuses méthodes basées sur l'apparence ont été proposées au fil des ans pour améliorer les performances de la reconnaissance des expressions du visage, la plupart des descripteurs ne sont généralement pas en mesure de fournir une analyse précise à la fois multi-échelle / multi-orientation et de gérer le problème de redondance efficacement.

Nous proposons donc dans ce chapitre d'utiliser les coefficients résultant des projections sur une base de polynômes pour la représentation de texture Pour extraire les caractéristiques faciales, nous proposons de calculer les coefficients issus de projections polynômiales sur chaque point d'intérêt du visage. Comme précédemment, deux modes de calcul sont disponibles: les coefficients peuvent être calculés soit sur des régions de texture, soit récupérées à partir d'une décomposition polynomiale multi-résolution.

Pour le premier mode - **SR_Poly**, le vecteur de caractéristique pour chaque point du visage est extrait d'un patch d'image de taille 19×19 pixels centré sur le point. Cette taille a été choisie pour être similaire à la taille calculée empiriquement pour l'approche en utilisant des histogrammes LBP. Étant donné que les coefficients polynomiaux fournissent

une représentation hiérarchique des structures de l'image, nous pouvons réduire leur nombre pour accélérer les calculs avec peu de perte d'efficacité.

Pour le deuxième mode -**MR_Poly**, nous utilisons une approche multi-résolution de 3 niveaux. Pour avoir une représentation similaire aux ondelettes de Gabor comme [ZLSA98], nous utilisons une base complète de taille 3×3. De cette façon, nous aurons une représentation avec 3 échelles et 9 orientations. Les régions autour de chaque point d'intérêt varient donc entre 81×81 et 3×3 pixels.

Les résultats expérimentaux obtenus sur deux bases d'images contenant des émotions comparés à ceux obtenus par trois méthodes de l'état de l'art confirment que notre approche fonctionne bien avec la reconnaissance de l'expression faciale, donnant des résultats de haute précision et des calculs efficaces lorsque les points clés du visage sont étiquetés manuellement ou calculés par un algorithme d'alignement.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

AAM Active Appearance Models

CDAAM Compressed Discriminative AAM

CLM Constrained Local Models

CPR Cascaded Pose Regression

FT-PAAM AAM retrieved from a multi-resolution polynomial decomposition of the full aligned texture

PAAM AAM calculated using polynomial projections on texture patches sampled around key landmarks

PDM Point Distribution Model

SVM Support Vector Machine

Chapter 1

Introduction

The face is a very rich source of information on non-verbal communication. Although a human observer is able to perceive naturally some of this information from visual observations, its analysis remains a very difficult task in computer vision. As one of the most active and visible research topics in pattern recognition, biometrics and image processing, facial analysis has been extensively studied in the past two decades due to its many application areas such as security (surveillance, biometrics), human-machine interaction, robotics, indexation, behavior analysis etc.

Face analysis research includes several themes : detection, tracking, localization, recognition, authentication, face synthesis. Algorithms used for face analysis face multiple challenges, including both intrinsic (pose or facial expression of the subject) and extrinsic parameters, such as partial occlusions or conditions of image acquisition (luminance problems, shadows).

During the course of this thesis, we will be mainly interested in polynomial modeling applied to face analysis and more specifically to deformable models : a set of methods that provide the abstract model or approximation of an object class. They model separately the variability in shape, texture or imaging conditions of the objects in the class (*e.g.* human faces), using a defined number of parameters. Polynomial representations have simple expression, allow to describe discrete sets of values (*ie* image pixels) by analytical functions and fit geometrical forms of images harmoniously. Slowly varying surfaces (like facial skin) in images are well represented by polynomials and their reconstruction quality is pleasant to the human eye.

The work presented here will primarily investigate the usage of polynomial representation applied to multiple face analysis steps, from face texture modeling and compression or facial keypoints detection to descriptors for facial expressions classification.

1.1 Motivation

The work presented in this thesis was motivated by a practical application for facial animation, and more generally the synthesis of human characters and scenes for the entertainment industry (video games, animated films).

This is usually done through automation or semi-automation of a part of the synthesis process based on a complex analysis of facial expressions and head movements. Traditionally, such animations are entirely produced by skilled actors on which are positioned markers. Although this type of animation gives the best quality results, its associated implementation process is slow and expensive because it requires a specific makeup and usually involves multiple cameras. Moreover, it is particularly inconvenient for the actors and for later video captures. Recent advances in image processing make possible the detection of facial features that can be exploited for automatic animation, without the use of markers. They provide animated meshes that can be injected into the synthesis and animation software, both in post-production and live, thus allowing to save time and providing significant shooting facilities.

Yet it is still a challenging problem in practice due to uncontrolled environment, occlusions and variations in pose, illumination, etc. Various methods have been proposed for facial features extraction, with different advantages and drawbacks in terms of robustness, computational cost and accuracy. Recent advances in image representation show that most low-level descriptors used in said methods rely on ill-defined frameworks.

A good example of low-level image descriptor is the Gabor space. Neurophysiological studies show evidence that the human visual system (HVS) is best modeled as a family of self-similar 2D Gabor functions [Dau85]. Like the Haar transform, the Gabor transform is considered as the mother wavelet in time-frequency analysis theory and is often used in facial analysis-related computer vision applications to create sparse object representations.

Despite their high accuracy, the use of Gabor filters in image processing is often criticized in terms of computational cost. Therefore we decided to study the case of another image representation having similar properties to the HVS, namely the orthogonal polynomials [Bla74]. Within the framework suggested by Blaivas, visual analysis in the retina can be regarded as a process of expansion in orthogonal polynomials basis. Motivated by this property, and by the results obtained concerning the use of orthogonal polynomials for human motion analysis in [KTAK10] we propose to use the coefficients derived from polynomial projection in several face analysis applications.

Therefore we will use bivariate orthogonal polynomials to construct 2D wavelet functions and to define a multiresolution wavelet-like image transform. We will see later that with respect to classic time-frequency representations, such as wavelets, polynomial basis

decompositions do not necessarily use a dyadic partition and are therefore more adaptable. The polynomial multi-resolution decomposition will allow to organize hierarchically the image information within the frequency domain. As a result, polynomial coefficients can be used as an efficient alternative to global or redundant texture representations such as Gabor Wavelets, without losing accuracy.

The purpose of the thesis is to study 2D polynomial modeling for image representations and see their impact in facial analysis applications (pose/landmark/expression detection for avatar animation, emotion detection, facial keypoints) in terms of robustness, accuracy and computational cost.

1.2 Thesis outline

This document is organized into three parts.

First, in chapter 2 a review of state of the art in face landmarking is presented. This review includes work on parametric and non parametric models for facial landmarks localization. This chapter is important to understand the motivation behind the work in this thesis. Chapter 3 begins with the presentation of a method to generate orthogonal polynomials bases. Follows the 2D polynomial image approximation that allow to obtain a null approximation error and the construction of a multiresolution piecewise polynomial decomposition. The chapter ends with the presentation of the strategies for polynomial coefficient selection in the approximation process.

Chapter 4 and 5 are dedicated to the integration of polynomial bases in deformable models and mainly the Active Appearance Models (AAM) algorithm. In chapter 4 we explore the way polynomial coefficients can be used for texture analysis and representation. First we detail the AAM framework and propose two different schemes to replace the original AAM texture representation model by approximating image structures with polynomial projections on an orthonormal basis. Next we extend an existing work applied to face alignment where wavelet coefficient subsets were modeled rather than pixel intensities by using subsets of polynomial coefficients. Deviating from the existing approach the approximation coefficients are included in a regression framework. In chapter 5 we will take an interest in the generative fitting, review in details the inverse compositional approach and see how the polynomials can be used to replace analytically the computation of gradients in the Gauss Newton and Newton descent gradient approach using projections on the first and second order basis polynomials.

The last part of this thesis have concentrated on the use of polynomial bases for keypoint detection and for facial expression recognition. Chapter 6 begins with the definition of the detection algorithm. First we show how to compute an image vector field of normals and then we present the selection of interesting points in a multi-scale and multi-resolution scheme. We evaluate next and compare to nine recent detectors our method on the Oxford dataset. We detail then how areas detected with our algorithm can be used in the AAM for a sparse texture model. Finally, in Chapter 7 we detail the way in which the results of the tracking algorithm can be exploited to the description of expressions. We first present a polynomial based texture representation model as a descriptor for facial expression information. We introduce two different modes for the descriptor calculation and compare in terms of computational efficiency the polynomial and Gabor transforms. We finish by describing the experimental results on two different databases.

1.3 Main contributions

The contributions of this thesis are consistent with the logical flow of the chapters. The first contributions rely to the discriminative fitting approach. We propose a new approach for texture representation in deformable models, and the inclusion of the polynomial coefficients in a regression framework to have a compressed polynomial texture model. The second contributions rely with the generative fitting approach. We propose to adapt the inverse compositional approach using polynomial bases both for the gradient descent algorithm as for the texture representation model. Finally, we investigate the use of polynomial bases for interesting points detection and facial expressions classification.

A more detailed description of the contributions is presented here:

- Since model fitting parameters are estimated by minimizing the sum of squared differences of texture values between observations and approximations, accuracy and robustness will rely heavily on choices regarding texture representation. In the polynomial texture representation for deformable models we use coefficients resulting from polynomial projections of pixel values for image approximation. By comparing our approach to PCA-based global analysis of raw pixel intensities one can usually find in AAM texture models we demonstrate it's ability to improve robustness against pose and facial expression changes.
- The state of the art AAM texture models explicitly model the value of every pixel covering an object. To avoid the excessive computational requirements for high

resolution images we propose to use a compressed polynomial texture model. This work is an extension of the work of Wolstenholme and Stegmann [SFC04] where Haar and CDF 9/7 wavelet coefficient subsets were modeled rather than pixel intensities. We demonstrate the straightforward integration of compressed polynomial coefficients in the texture model of an AAM and introduce a framework that incorporates polynomial compression into a cascaded regression.

- Considering the polynomial approximation equivalent to a filter bank, we show first how the polynomial coefficients can be used in a gradients descent algorithm. Next we reformulate the inverse compositional algorithm to entertain fitting across multiple polynomial filter responses and show how using polynomial bases in the Gauss Newton and Newton gradient descent algorithm limits the computation error and induces better alignment results.

- We describe an algorithm for points of interest detection that works not only on grayscale images (as the majority of recent detectors) but also on color images. This algorithm offers the possibility to use different types of bases (and therefore use different types of smoothing in the construction of scale pyramids) while creating the multivariate basis and allows to select the number of detected features using simple thresholds. The facial regions chosen by our algorithm are also included in an AAM texture model.

- Finally we propose to use polynomial bases for feature extraction within a system of facial expression recognition. Two different modes for the description are presented, using the single or multi resolution approach polynomial projections. In the single resolution approach the coefficients provide a hierarchical representation of image structures, therefore their number is reduced to speed-up the computations with little efficiency loss. The multi resolution approach is showed to be more compact than a Gabor wavelet representation, thus allowing the disappearance of most sampling problems, such as the trade-off between orientation sampling and spatial sampling.

- The performance of all the proposed method and algorithms is evaluated and compared on several databases, demonstrating their accuracy.

State of Art

Chapter 2

Recent advances in face landmarking

2.1 Introduction

A landmark represents a distinguishable point present in most of the images under consideration, for example, the location of the left eye pupil. Facial landmark estimation seeks to automatically locate predefined facial landmarks in face images (Fig. 2.1). It is an important research area in computer vision in part because digital face portraits are ubiquitous. Accurately modeling human faces is key for a number of visual tasks such as facial recognition [ND09, WFKVDM97], face reconstruction [KSS11], expression recognition [Bet12, LMH+06], facial animation [CWLZ13] or biomedical applications [BML+01] to name just a few.

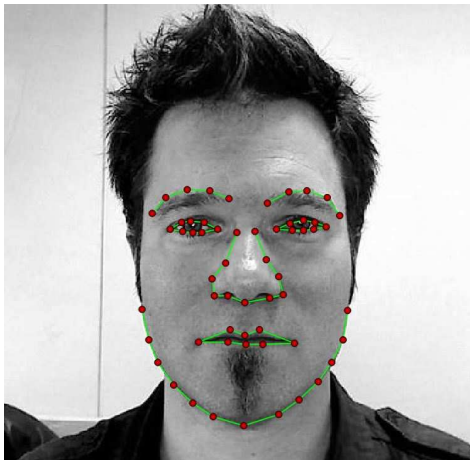


Figure 2.1 A face with correctly positioned landmarks

Robust facial landmark estimation is very challenging in practice, due to a variety of factors such as acquisition cameras, physiognomies, illumination effects, occlusions or poses.

Furthermore accurate and precise landmarking remains a difficult problem since, except for a few, the landmarks do not necessarily correspond to high-gradient or other salient points. Hence, low-level image processing tools remain inadequate to detect them, and recourse has to be made to higher order face shape information.

Early work on the facial landmark localization [EFM09] often addressed the problem as a particular case of the object part detection problem. However, general detection methods are not adapted to detect facial landmarks because as mentioned before few salient markings (eg, centers for eyes, lips) can be characterized reliably by their image appearances. Therefore, shape constraints or support neighboring areas are essential for augmenting weak local detectors. According to the type of constraints imposed previous work can be classified into two groups: the **parametric methods** and **non parametric methods**.

The majority of methods described lower depends on a good face initialization. A popular strategy, even for recent approaches (e.g., [AZCP13], [CWWS14], [CCTC09], [XDlT13] to name just a few), is to first detect the face (i.e., using [VJ04]), and then fit a mean face shape (where the shape is defined by the facial landmarks) to the detection window. However, for extreme poses and some expressions, traditional face detectors (e.g., [VJ04]) may fail, or the true shape of the face inside the detection window will differ significantly from the initial shape, making a good initialization unlikely.

Part-based models [FGMR10], [YR11] can be used to address the initialization problem, but learning an accurate part graph parameterization and inferring part labels from the graph can be challenging. Recent works [YHZ+13], [ZR12] simplify the graph structure to a tree and produce impressive results.

Though it is possible to build separate Active Appearance Models or Active Shape Models to handle pose variation (view-based models), as carried out in [CWWT02], [RGP+99] [ZA06] and [MBN13], respectively, the fact that they require very accurate initialization decreases their effectiveness, especially on real-world images, where the simultaneous effects of pose (yaw, pitch, and roll) and facial occlusions can decrease their accuracy. Thus, there has been a recent increase in literature dealing with the automatic landmarking of non-frontal faces using various unique approaches. Everingham et al. [ESZ06] used a generative model of facial feature positions (modeled jointly using a mixture of Gaussian trees) and a discriminative model of feature appearance (modeled using a variant of AdaBoost and "Haar-like" image features [VJ01]) to localize a set of 9 facial landmarks.

2.2 Parametric models

Traditionally facial landmarking has been carried using deformable template (parametric) based models that can roughly be divided into two main categories: (a) Holistic Models that use the holistic texture-based facial representations; and (b) Part Based Models that use the local image patches around the landmark points. Notable examples of the first category are Active Appearance Models (AAMs) [CET01a],[TAiMZP13] and 3D deformable models [BV03]. The second category includes models such as Active Shape Models (ASMs) [CTCG95], Constrained Local Models (CLM) [SLC11] and the tree-based pictorial structures [ZR12]. We will not discuss here the 3D deformable models.

2.2.1 Active Shape Models

The Active Shape Model (ASM) was introduced by Cootes et al. [CTCG95] as a method of fitting a set of local feature detectors to an object and simultaneously taking into account global shape considerations. The allowable shape deformations are learnt from a manually labelled training set to produce a linear shape model with the following form:

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s}\mathbf{b_s} \tag{2.1}$$

where $\bar{\mathbf{s}}$ is the mean shape, $\mathbf{P_s}$ is a set of orthogonal modes of variation and $\mathbf{b_s}$ is a set of shape parameters. An illustration of a statistical distribution of facial feature points is represented in Figure 2.2. There are 600 shapes (smaller dot points in black) normalized by Procrustes analysis. The larger dot points in red indicate the mean shape of all shapes.

Various shapes can be generated with Equation 2.2 by varying the vector parameter $\mathbf{b_s}$. By keeping the elements of $\mathbf{b_s}$ within limits (determined during model building) the generated face shapes are lifelike. Conversely, given a new shape $\tilde{\mathbf{s}}$, the parameter $\mathbf{b}$ that allows to produce $\tilde{\mathbf{s}}$ given a model shape $\bar{\mathbf{s}}$ can be calculated.

Cootes and Taylor [CT+04] describe an iterative algorithm that gives the $\mathbf{b_s}$ and $\mathbf{T}$ that minimizes

$$distance(\tilde{\mathbf{s}}, \mathbf{T}(\bar{\mathbf{s}} + \mathbf{P_s}\mathbf{b_s})) \tag{2.2}$$

where $\mathbf{T}$ is a similarity transform that maps the model space into the image space.

Many modifications to the classical ASM have been proposed over the years, such as in [MN08], [SS09], that have mainly focused on **developing better local texture models**, however they still remain susceptible to occlusions, the problem of local-minima, and are very dependent on good initialization Other improvements focus on **the local detectors**. For

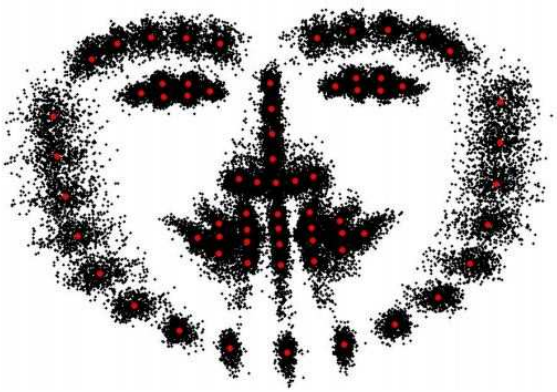


Figure 2.2 Statistical distribution of facial feature points. Figure taken from [WGTL14]

example, Boosted Regression Active Shape Models [CC07] use boosting to predict a new location for each point, given the patch around the current position.

Among the methods focusing on a more **robust global shape prior**, Everingham et al. [ESZ06] model the face configuration using pictorial structures [FH05], a hierarchical version of which was used in [RBDlT+11]. Valstar et al. [VMBP10] combine SVM regression for estimating the feature point's location with conditional Markov random fields to keep the estimates globally consistent. They also take advantage of facial feature points whose position is less sensitive to facial expressions; they thus start by localizing such stable points first and then find the additional points after a registration step. The whole process takes around 50 seconds per image. Very recently, Amberg and Vetter [AV11] proposed to run detectors over the whole image and then find the optimal set of detections using Branch & Bound; however, they only show results for high-quality images and need over one second to process one image.

ASMs belong to a class of methods that can be broadly referred to as Constrained Local Models (CLMs) [CC06], [CC08], [SLC11] (see 2.2.3).

2.2.2 Active Appearance Models

An AAM is a statistical model introduced by Cootes et al [CET01a] which describes shape and texture variabilities of an object learned from a training set comprising different views of the object.

Similar to ASMs, active appearance models are created from manually annotated data with key landmarks points images and where the variations between the positions of points - the shape **x** and the pattern of intensities or colors across an image patch - the texture **g**, are learned by principal component analysis. Any example can be approximated using :

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s b_s} \qquad \mathbf{a} = \bar{\mathbf{a}} + \mathbf{P_a b_a} \tag{2.3}$$

where $\bar{\mathbf{s}}$ and $\bar{\mathbf{a}}$ are respectively the mean shape and texture, $\mathbf{P_s}, \mathbf{P_a}$ are matrices describing the modes of variation derived from the training set. A third PCA is then performed on a concatenated shape and texture parameters $\mathbf{b}$, to obtain a combined model vector $\mathbf{c}$:

$$\mathbf{b} = \mathbf{Qc} \tag{2.4}$$

From the combined appearance model vector $\mathbf{c}$, a new instance of shape and texture can be generated:

$$\mathbf{s_{model}} = \bar{\mathbf{s}} + \mathbf{P_s W_s Q_s c} \qquad \mathbf{a_{model}} = \bar{\mathbf{a}} + \mathbf{P_a Q_a c} \tag{2.5}$$

where $\mathbf{W_s}$ is a diagonal matrix of weights for each shape parameters and

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q_s} \\ \mathbf{Q_a} \end{pmatrix} \tag{2.6}$$

Fig. 2.3 shows the effect of varying the first four parameters from $\mathbf{c}$, showing changes in identity, pose, and expression. Note the correlation between shape and intensity variation.

The goal of the active appearance model is to find the appearance parameters giving the best match to an unseen image. Both ASMs and AAMs build shape models (also referred

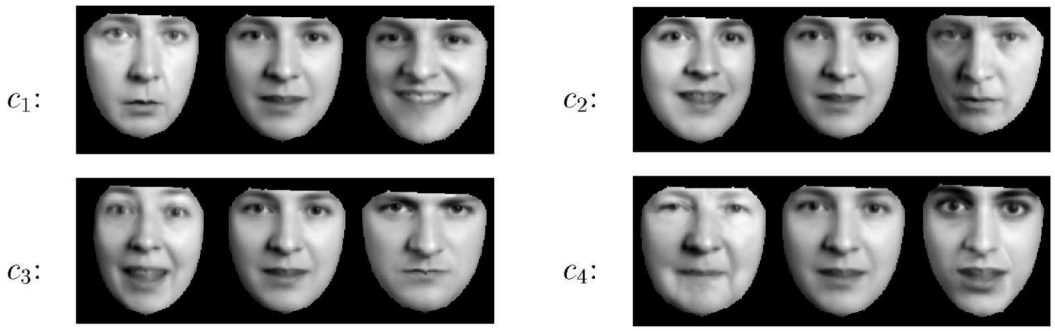


Figure 2.3 Effect of varying first four facial appearance model parameters, $c_1 - c_4$ by ± 3 standard deviations from the mean. Taken from [CET01a].

to as Point Distribution Models (PDMs)), that model the shape of a typical face that is represented by a set of constituent landmarks, and texture models of what the region enclosed by these landmarks looks like. The difference between the two is that ASMs build local texture models of what small 1D or 2D regions around each of landmarks look like, while AAMs build global texture models of the entire convex hull bounded by the landmarks.

The two main assumptions behind AAMs are that (1) for every test (unseen) image there exists a test shape and set of texture weights for which the test shape can be warped onto the reference frame and expressed as a linear combination of the shape-free training textures and (2) the test shape can be written as a linear combination of the training shapes.

Defining a linear statistical model of texture that explains variations in identity, expressions, pose and illumination, is a very challenging task, especially in the intensity domain. Furthermore, the large variation in facial appearance makes it very difficult to perform regression from texture differences to shape parameters. That's why numerous extensions of standard AAMs have been proposed to improve their fitting quality.

The majority of AAM extensions can loosely be categorized based on how they tackle the problem, with the most common strategies being: (1) **improvement of the actual fitting procedure** by changing the factors involved in the optimization (e.g. [GMB05], [CT06], [MB04],[GMB06]); and (2) **usage of more robust feature representations**, e.g. to obtain invariance with respect to occlusions [TAiMZP13], illumination [NSL11], or non-linear shape deformations [HM09].

One of the main disadvantages of the algorithm, as for instance stated in [TAiMZP13], [GMB05], [CT06],[Liu10], [PPB08], [SK09] is their **weak generalization** ability when learned with only a few training examples that do not cover the complete range of possible variations in the data. To overcome this problem Zhao et al. [ZSCC13] proposed computing a separate AAM for each test face using k-nearest neighbor training faces (w.r.t. the test face) rather than all training faces. Using k-NN exemplars is an important part of the approach of [BJKK11] [ZBL13],[SLBW13].

To align unknown faces in unknown poses and illuminations, [SLGBG09] proposed to use specific transformation of the active model texture in an oriented map, which changes the AAM normalization process and to do the research in a set of different precomputed models related to the most adapted AAM for an unknown face.

The classic AAM approach is computationally expensive and sensitive to the initialization due to the involved gradient descent based optimization[YHL+03, CILS12, XDlT13, CWWS14].

Recently, Tzimiropoulos and Pantic [TP13] proposed new optimizations for fast and accurate AAM fitting and demonstrated better fitting results on unseen images with a large range of pose variation using a more unconstrained training set drawn from the Labeled Face Parts in the Wild (LFPW) dataset [BJKK11].

Approaches which increase the **expressiveness** of an AAM are very rare. One example is the Online Appearance Model [SK09], where the texture component is constantly being updated via incremental PCA during model fitting to account for illumination changes.

Similarly, Adaptive AAMs [Liu10] feature a generic and a subject-specific texture component, where the latter is again being updated during fitting. However, both methods add knowledge to the model only at fitting time and, both approaches update only the texture component, which, for instance, excludes the possibility to add new facial expressions to existing AAMs.

Although the well known inverse compositional ICIA algorithm [MB04] has been criticized for its inability to perform well under generic fitting scenarios, i.e. to fit images of unseen identities, the algorithm is very popular, mainly because of its extremely low computational complexity, and methodologies such as [PM08][ABV09], which can provide near real-time fitting, has not received much attention. It has been demonstrated that ASMs are more suited to the task of precise facial landmarking than AAMs [CET01a], [SS12], [CWWT02], [CT$^+$04], [BM04] [CET99], as AAMs are generative, global texture based approaches and are more easily affected by variations in illumination and the presence of occlusions.

Compared to ASMs, AAMs generalize poorly to unseen faces, however for tasks where generalizing across people is not necessary and one has access to several training images of an individual, AAMs work very well and are able to learn a holistic representation of the face. Therefore active appearance models, are useful for many tasks like face inpainting, detecting and removing occlusion, face identification, face animation.

2.2.3 Part based models

The part based models [ZR12] ,[CC08], [SLC11], [BJKK11], [CILS12] perform face alignment by maximizing a posterior probability of part locations given the image and then fuse the probabilities of all the parts together enforced by a global shape model, e.g. enhanced ASM [CC08], [SLC11] or pictorial structures [ZR12], to generate the final result.

The main advantages of part-based models [SLC11],[ZR12], [AZCP13] (i.e. models which do not define a complete holistic texture model of the object) are a natural handling of partial occlusions (since they only model certain parts of the object) and, most importantly, the fact that they are optimized only with respect to shape (they do not define parametric models of texture). Notable examples include Constrained Local Models (CLMs) [SLC11] and the tree-based model of [ZR12] (which can be also used for object detection). More recently, Asthana et al. [AZCP13] proposed a robust discriminative framework for fitting CLMs which achieved state-of-the-art results in the problem of facial alignment "in the wild".

CLM

CLMs [CC08], build local models of texture variation around landmarks (sometimes referred to as "patch experts" and allow landmarks to drift into the locations that best match training data using these patch experts. This is similar to the AAM; however, the texture sampling method is different. The shape is then regularized using the shape model to generate a plausible set of final landmark locations. Unlike AAM which tries to approximate the raw image pixels directly, the constrained local models [CC08] employ an extended appearance model to generate the feature templates of the parts, which obtains improved robustness and accuracy. The search algorithm is presented in Fig.2.4.

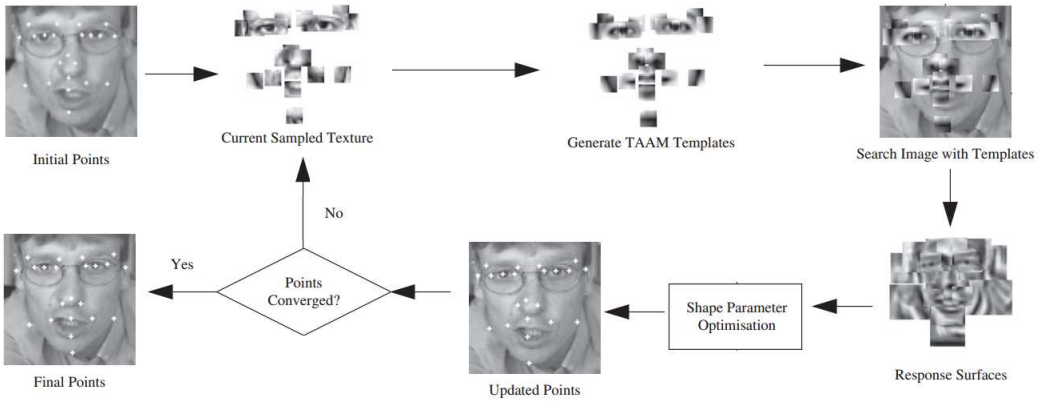


Figure 2.4 Constrained Local Model (CLM) search algorithm. Taken from [CC08]

The local appearance models are more robust to a range of challenges including occlusion and global illumination changes, but CLMs still rely on parametric shape models for regularization, which may not generalize well to a broad range of poses.

Saragih et al. [SLC11] proposed to use linear SVMs over power normalized image patches to discriminate aligned from misaligned mesh vertex coordinates. Composing the SVM classification score with a sigmoid function generates a likelihood map over the vertices within a local search region around its current estimate. This allows a Bayesian treatment of the alignment problem. Asthana et al. [AZCP13] developed a discriminative regression based approach for the CLM framework that they referred to as Discriminative Response Map Fitting (DRMF). DRMF represents the response maps around landmarks using a small set of parameters and uses regression techniques to learn functions to obtain shape parameter updates from response maps.

Tree based models

In their recent seminal work, Zhu and Ramanan [ZR12] proposed an elegant framework that built on the previously developed idea of using mixtures of Deformable Part Models (DPMs) for object detection [FGMR10] to simultaneously detect faces, localize a dense set of landmarks, and provide a course estimate of facial pose (yaw) in challenging images. Their approach used a mixture of trees with a shared pool of parts V to model each facial landmark. Global mixtures were used to capture changes in facial shapes across pose and the tree-structured models were optimized quickly and effectively using dynamic programming. The approach is quite effective at handling a wide range of yaw variation but does not account for excessive in-plane rotation of faces or large occlusion levels.

In their tree structured part model each tree is written $T_m = (V_m, E_m)$ as a linearly-parameterized, tree-structured pictorial structure, where m indicates a mixture and $V_m \subseteq V$. Taking an image I , and the pixel location of part i as $l_i = (x_i, y_i)$ a configuration of parts $L = \{l_i : i \in V\}$ is scored as:

$$S(I,L,m) = App_m(I,L) + Shape_m(L) + \alpha^m \tag{2.7}$$

$$App_m(I,L) = \sum_{i \in V_m} w_i^m \cdot \phi(I,l_i) \tag{2.8}$$

$$Shape_m(L) = \sum_{ij \in E_m} a_{ij}^m dx^2 + b_{ij}^m dx + c_{ij}^m dy^2 + d_{ij}^m dy \tag{2.9}$$

Equation 2.8 sums the appearance evidence for placing a template w_i^m for part i, tuned for mixture m, at location li . $\phi(I,l_i)$ represent the feature vector (e.g., HoG descriptor) extracted from pixel location l_i in image I. Equation. 2.9 scores the mixture-specific spatial arrangement of parts L, where $dx = x_i - x_j$ and $d_y = y_i - y_j$ are the displacement of the ith part relative to the jth part. Each term in the sum can be interpreted as a spring that introduces spatial constraints between a pair of parts, where the parameters (a,b,c,d) specify the rest location and rigidity of each spring. Finally, the last term α^m is a scalar bias or "prior" associated with viewpoint mixture m.

A comparison of the learned shape models with those trained generatively with maximum likelihood is presented in Fig. 2.5. Tree based SVM captures much of the relevant elastic deformation, but produces some unnatural deformations because it lacks loopy spatial constraints.

Since pose is part of estimation, the algorithm practically works as a multiview algorithm. In contrast to [VMBP10], [MVBP13] where local and global information are invoked in succession, this algorithm is shape driven, and local and global information are merged

right from beginning. This is implemented by considering several (30 to 60) local patches that are connected as a tree, which collectively describe the landmark related region of the face; in other words, the patch-based face graph models the ROI of the detected face and incorporates its pose and landmark information. This approach is an adaptation of the idea of tree-structured pictorial structures [FGMR10].

Recently, Ghiasi and Fowlkes [GF14] built on this work and proposed a hierarchical deformable part model for face detection and landmark localization to explicitly model the occlusion of parts and hence achieved more accurate results on challenging occluded images in the wild.

Tree-structured pictorial structures have also been successfully applied to face recognition by Everingham et al. [CWWS14], where the local appearance of each landmark is learned by a variation of Adaboost algorithm with Haar-like features [VJ04]. Similarly, Uricar et al. [UFH12], inspired by pictorial structures, jointly optimize appearance similarity and deformation cost with a parameterized scoring function where the parameters are learned from manually annotated instances using the structured output SVM classifier.

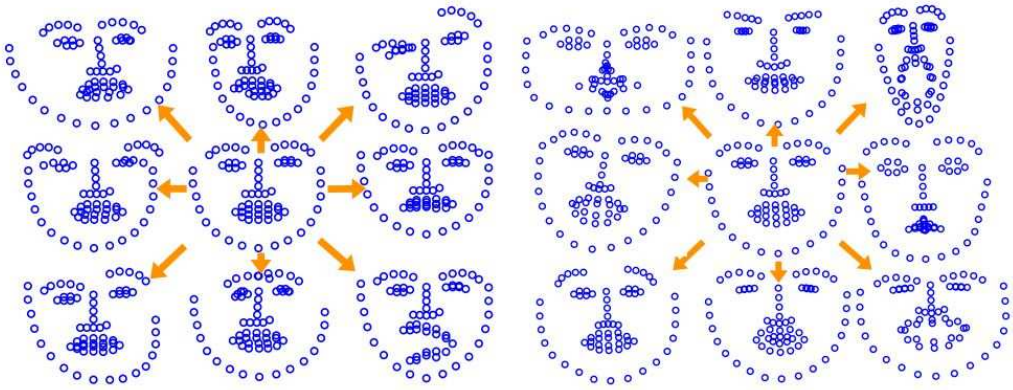


Figure 2.5 Mean shape and deformable models for tree based SVM (left) and AAM (right). Taken from [ZR12]

2.3 Non parametric models

Despite the success of parametric shape models, the model flexibility (e.g., PCA dimension) is often heuristically determined. Furthermore, using a fixed shape model in an iterative alignment process (as most methods do) may also be suboptimal. This is the reason, recently non parametric methods have emerged.

2.3.1 Shape regression

Recently, a variety of approaches [CWWS14],[BAPD13],[YLYL13], [KJ14] that can be broadly grouped under the category of shape regression based approaches have emerged. Regression based methods can achieve accurate results at great speed and have thus become quite popular. All these methods are improved variants of the original approach called Cascaded Pose Regression (CPR) and introduced by Dollar et al. [DWP10].

CPR is formed by a cascade of T regressors $R^{1..T}$ that start from a raw initial shape guess S^0 and progressively refine estimation, outputting final shape estimation S^T . Shape S is represented as a series of P part locations $S_p = [x_p; y_p], p \in 1..P$. At each iteration, regressors R^t produce an update ∂S, which is then combined with previous iteration's estimate S^{t-1} to form a new shape.

The training procedure for a CPR is shown in Alg. 1. During learning , each regressor R^t is trained to attempt to minimize the difference between the true shape and the shape estimate of the previous iteration S^{t-1} using landmark-indexed features h_t. For simplicity, we use the notion of "landmark-indexed" feature instead of "pose indexed features" used by the authors of [DWP10]. S_0 is the single pose estimate that gives the lowest training error without relying on any component regressors. The available features depend on the current shape estimate and therefore change in every iteration of the algorithm; such features are known as landmark-indexed features. The key to CPR lies on computing robust landmark-indexed features and training regressors able to progressively reduce the estimation error at each iteration.

Algorithm 1 Training for cascaded Pose Regression (taken from [DWP10])

Input: Data (I_i, S_i) for $i = 1..N$
1: $S_0 = \arg\min_S \sum_i d(S, S_i)$
2: $S_i^0 = S^0$ for $i = 1...N$
3: **for** $t = 1$ to T **do**
4: $\quad x_i = h^t(S^{t-1}, I_i)$
5: $\quad \tilde{S} = \bar{S}_i^{t-1} \circ S_i$
6: $\quad R_t = \arg\min_R \sum_i d(R(x_i), \tilde{S}_i)$
7: $\quad S_i^t = S_i^{t-1} \circ R^t(x_i)$
8: $\quad \varepsilon_t = \sum_i d(S_i^t, S_i) / \sum_i d(S_i^{t-1}, S_i)$
9: $\quad$ If $\varepsilon_t \geq 1$ stop
10: **end for**
11: Output $R = (R^1, ..., R^T)$

After the training step, given an input shape S_0, the regressor $R(S_0; I)$ is evaluated by computing: $S^t = S^{t-1} \circ S_\delta$ from $t = 1...T$ and finally outputting S^T. (see Algorithm 2)

Algorithm 2 Evaluation of cascaded Pose Regression (taken from [DWP10])

Input: Image I, initial shape S^0

1: **for** $t = 1$ to T **do**
2: $x = h^t(S^{t-1}, I)$ // compute features
3: $S_\delta = R^t(x)$ // evaluate regressor
4: $S^t = S^{t-1} \circ S_\delta$ // update S^t
5: **end for**
6: Output S^T

The algorithm uses 5 depth random fern regressors as regressors R^t and landmark-indexed control point features. Each control point feature is computed as the difference of two image pixels at predefined image locations. In Fig.2.6 the yellow crosses represent the coordinate system defined by the current estimate of the pose of the object (which does not have to be centered on the object). The colored arrows show control points defined relative to the pose coordinates.

Each fern selects which 5 features to use from a large pool of F features via either a random-step optimization or a correlation-based evaluation which is faster and improves performance.

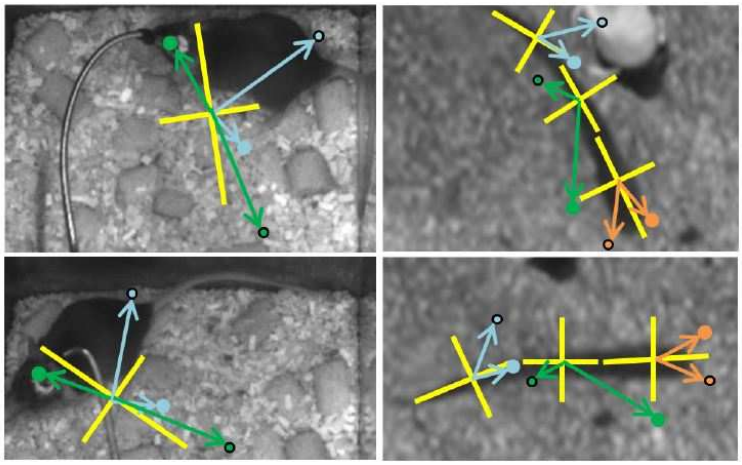


Figure 2.6 Landmark-indexed features. Left: Mice described by a 1-part pose model. Right: 3-part pose model of zebra fish. Taken from [DWP10]

Cao et al. [CWWS14] proposed a number of improvements over CPR. They point out that local evidence is sufficiently strong only for a few prominent landmarks, but otherwise most others are not salient enough and cannot be reliably characterized by their image appearance, and therefore shape constraint is essential. Their method is regression based where the shape constraint is realized in a nonparametric manner. Their nonparametric

approach is based on the fact that the regressed shape is a linear combination of all training shapes. An interesting aspect is that instead of using the regressors in parallel and fusing their result as in [KSYY10] the authors use sequential regressors, where each one in the sequence uses the image information and the shape estimated from the previous stage of regression. Furthermore,the regressed shape is always constrained to reside in the linear subspace constructed by all training shapes. This guarantees the plausibility of the shape as well as global consistency.

The two-tier approach of Valstar et al. [VMBP10] uses in the first level surrounding image information to predict landmark location via support vector regression (SVR), and in the second level, the global shape information via a Markov Network. The regressor simplifies the landmark search in contrast to exhaustive sliding-window search with a template window.

To explicitly deal with occluded faces and provide feedback on which landmarks were occluded Burgos-Artizzu et al. [BAPD13] proposed the Robust Cascaded Pose Regression (RCPR) algorithm. They incorporated occlusion directly into the learning stage, using facial images that were both manually annotated and provided with occlusion labels, to improve shape estimation.

Other regression based face alignment approaches are used in [CC07], [VMBP10], [CILS12], [SG07] and [DWP10]. The distinctions among these methods mainly lie in the employed learning algorithm (e.g. boosting [CC07], random forest [CILS12], or non-linear least squares [XDlT13]) and the adopted features (e.g. Haar wavelets [CC07], random ferns [DWP10], or SIFT [XDlT13]).

2.3.2 Other methods

Random forests and ferns

Dantone et al. [DGFVG12] propose pose-dependent landmark localization scheme that is achieved by conditional random forests. While regression forests try to learn the probability over the parameter space from all face images in the training set, conditional regression forests learn instead several conditional probabilities over the parameter space, and thus can deal with facial variations in appearance and shape. The head pose is quantized into five segments of "left profile, left, front, right and right profile" faces and specific random forests are trained. The local properties of a patch is described both by texture and by 2D displacement vectors that are defined from the centroid of each patch to the remaining ones. Specifically, texture is described by Gabor filter responses in addition to normalized gray values in order to cope with illumination changes.

Training of conditional random forests is very similar to random forests; the main difference is that the probability of assigning a patch to a class is conditioned on the given head pose. This approach is able to deliver located landmarks in a query image at real-time speed.

Bayesian approach

Belhumeur et al. [BJKK11] use in an innovative manner a fully Bayesian approach to deduce landmark positions from local evidences. An interesting aspect of their work is that these evidences, that is, the local detector outputs are collected from a cohort of exemplars (sample faces with annotated landmarks), which thus provide non-parametrically the global model information.

In their recent work, [AiMZ] propose a Bayesian formulation of AAMs. To this end, they use a simple probabilistic model for texture generation assuming both Gaussian noise and a Gaussian prior over a latent texture space. The shape parameters are retrieved by formulating a novel cost function obtained by marginalizing out the latent texture space. This results in a fast implementation when compared to other simultaneous algorithms for fitting AAMs, mainly due to the removal of the calculation of texture parameters.

Semi-supervised learning

Tong et al. [TLWT12] address the tedious and often imperfect task of manual landmark labeling, and suggest a scheme to partly automate it. In their method, a negligible percentage (e.g., 3%) of faces need to be hand labeled, while the rest of the faces are automatically marked. This is realized by propagating the landmarking information of the few exemplars to the whole set. The learning is based on the minimization of the pairwise pixel differences resulting in two error terms: the penalty in one term controls the warping of each un-marked image toward all other un-marked images, so that they become more alike irrespective of the content. The penalty in the other term controls the warping of un-marked images toward marked images, and it is here that the physical meaning of the content is imposed. The warping function itself can be a global affine warp for the whole face, or a piecewise affine warp to model a non-rigid transformation.

Multi-kernel SVM

Rapp et al. [RSBP11] introduce a multi-resolution framework where low resolution patches carry the global information of the face and give a coarse but robust detection of the desired landmark and high resolution patches, using local details, refine this location. This

process is combined with a bootstrap process and a statistical validation, both improving the system robustness. Combining independent point detection and prior knowledge on the point distribution, the proposed detector is robust to variable lighting conditions and facial expressions.

2.4 Data description and experimental design

2.4.1 Data description

To evaluate the performance of our algorithms, we have carried out matching experiments on face images from the IMM [NLSS04], the CMU Multi-PIE [SBB03], MUG [APD10a] and Cohn Kanade [LCK+10] databases.

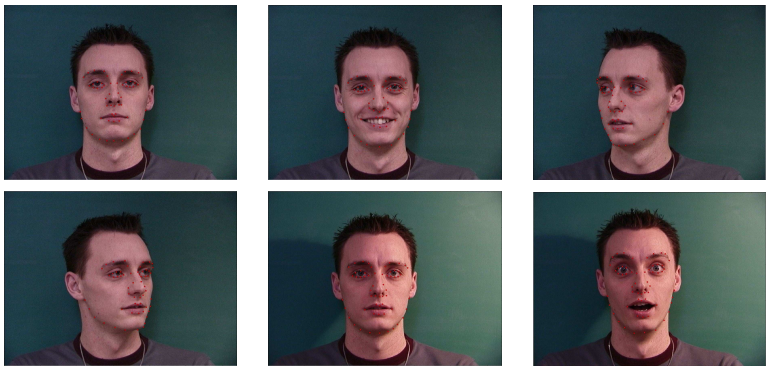


Figure 2.7 Example images from IMM database

The IMM data set contains 240 images annotated with 58 landmarks of 40 subjects,all without glasses, each of which having 6 different orientations and several facial expressions. It includes images of frontal and 30° rotated faces , under neutral; happy and an arbitrary expression taken under diffuse light or using a spot light added at the person's left side (see Fig. 2.7).

The MUG database consists of image sequences of 86 subjects performing frontal facial expressions, out of which 401 images of 26 subjects are manually annotated with 80 landmarks. (see Fig. 2.8).

CMU Multi-PIE face database contains images of 337 people imaged across different poses, under 19 different illumination conditions and while displaying a range of facial expressions. Although images in Multi-PIE are recorded under different in-door controlled lights, in our experiments, we only use images of Multi-PIE taken under frontal pose. We select a subset of 40 individuals images with various facial expressions that are hand annotated with 68 points. (see Fig. 2.9).

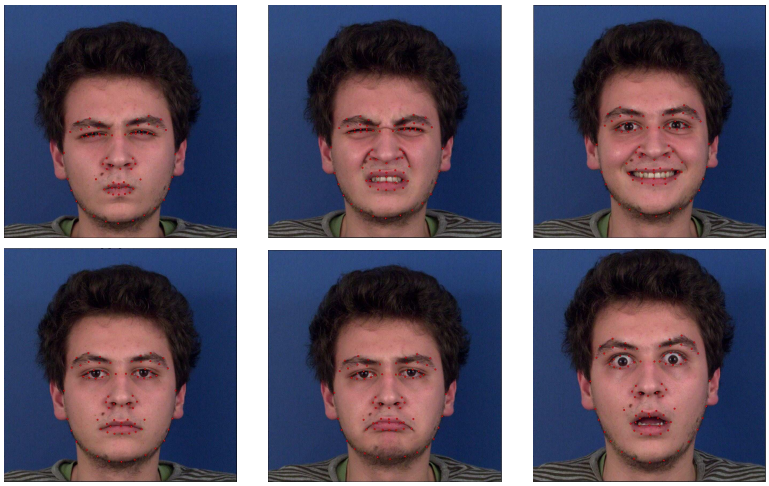


Figure 2.8 Example images from MUG database

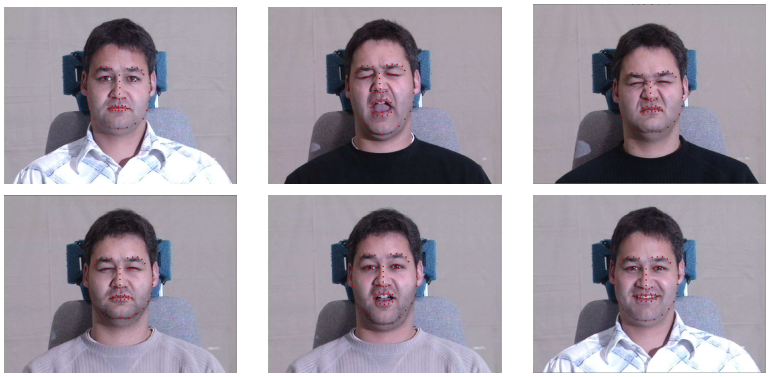


Figure 2.9 Example images from CMU Multi-PIE database

Figure 2.10 Example images from Cohn-Kanade dataset

The Cohn-Kanade database consists of expression sequences of 210 adults, annotated with 68 points, starting from a neutral expression and ending in the peak of the facial

expression. Participants were instructed by an experimenter to perform a series of 23 facial displays, six of which were prototypical emotions including angry, disgust, fear, joy, sadness and surprise. We use a subset of 115 subjects for our experiments. Only the first (neutral) and final image (the prototypical expression) of each of the selected sequences are considered for our training and testing. (see Fig. 2.10).

We have to highlight that all the algorithms have been trained and tested on the same data and using the same features

2.4.2 Measures to evaluate fitting performance

We report the performance of AAMs using two very popular error measures. The first criteria for success is the distance of the points computed using automated methods compared to manually labelled ground truth. The distance metric is shown in Equation 2.10:

$$m_e = \frac{1}{ns}\sum_{i=1}^{n} d_i \tag{2.10}$$

Here d_i are the Euclidean point-to-point errors for each individual feature location and s is the ground truth inter-ocular distance between the left and right eye pupils. According to the database, n is different as only the internal feature locations around the eyes, nose, brows and mouth are used to compute the distance measure. The feature points on the edge of the face are ignored for evaluation purposes.

The second measure is the point-to-point error defined as the Euclidean distance between the estimated landmarks $\mathbf{x}$ and the hand labelled landmarks $\mathbf{x}_{hl}$. The greater the error is, the worse the fitting is. The distance metric is shown in Equation 2.11.

$$E_{pt-pt}(\mathbf{x}_{hl}, \mathbf{x}) = \frac{1}{ns}\sum_{i=1}^{n} \sqrt{\left(x_i - x_{hl,i}\right)^2 + \left(y_i - y_{hl,i}\right)^2} \tag{2.11}$$

Here n is the number of keypoints that constitute the model and s the inter-ocular distance between the left and right eye pupils of the mean shape calculated during model building. For this error measure, we also produced the cumulative curve corresponding to the percentage of test images for which the error was less than a specific value.

We have presented in this chapter a review of state of the art in face landmarking. This review includes work on parametric and non parametric models for facial landmarks localization. We have also presented the databases and the measures used for experimental tests in order to evaluate the fitting performance of our algorithms. Since the purpose of

the thesis is to study 2D polynomial modeling for image representations the next chapter is dedicated to the presentation of 2D polynomial bases, 2D polynomial image approximation and the strategies used for polynomial coefficient selection in the approximation process.

Chapter 3

Image analysis with polynomials

The purpose of this chapter is to study the 2D polynomial image representation and see its impact in facial analysis applications. Encouraged by the results of the use of polynomial bases to model vector fields and to analyze simple face movements we wanted to extend the research to landmark/expression detection for avatar animation, person recognition and emotion detection and see its impact in terms of robustness, accuracy and computational cost.

Polynomial representations are similar to complete wavelet packet decompositions for a defined scale. Such descriptions have been used for the characterization and representation of handwritten mathematical symbols [CW07], the analysis of vowels and consonants in spectral frequency for speech recognition [EAS11], or the generation of linear phase two-dimensional FIR digital filter functions [CP12]. Their use in image representation has also been demonstrated in [Sad96, EUL86], while recent articles have studied the use of the discrete polynomial transforms for image coding [Mar06, KK09] or rotation invariant image texture image retrieval [K+12]. In [AC12], Carré and Augereau proposed a multi-scale hypercomplex 2D polynomial transform for color images based on quaternionic polynomials.

3.1 Complete bases

Let a Real Bivariate Polynomial of degree d be the function of $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}) \in \mathbb{R}^2$ defined as:

$$P(\mathbf{x}) = \sum_{\substack{(d_1,d_2)\in[0;d]^2 \\ d_1+d_2\leq d}} a_{d_1,d_2} x_1^{d_1} x_2^{d_2} \tag{3.1}$$

where $d_1 \in \mathbb{N}^+$ and $d_2 \in \mathbb{N}^+$ are the degrees of variables x_1, x_2 and the $\{a_{d_1,d_2}\} \in \mathbb{R}$ are the coefficients of the polynomials.

Considering a finite set of pairs $D = \{(d_1, d_2)\} \subset \mathbb{N}^2$, we represent by $\mathbb{E}_D$ the space of all real bivariate polynomials such as $a_{d_1,d_2} \equiv 0$ if $((d_1, d_2) \notin D)$ and by $\mathscr{K}_D$ the subset of real monomials:

$$\mathscr{K}_D = \left\{ K_{d_1,d_2}(\mathbf{x}) = x_1^{d_1} x_2^{d_2} \right\}_{(d_1,d_2)\in D} \tag{3.2}$$

Obviously $\mathscr{K}_D$ satisfies the linear independence and spanning conditions and so, $\mathscr{K}_D$ is a basis of $\mathbb{E}_D$, the canonical basis. In image analysis, we look for bases with suitable properties such as orthogonality or normality. So, to construct a discrete orthonormal real bivariate polynomial finite basis we first have to consider the underlying discrete domain:

$$\Omega = \left\{ \mathbf{x}_{(u,v)} = \left(x_{1,(u,v)}, x_{2,(u,v)} \right) \right\}_{(u,v)\in D_1} \tag{3.3}$$

where D_1 represents the set of pairs associated to Ω. To discretize the image domain two methods of collocation can be used, the first, the most classic is the uniform collocation (equation 3.4) and the second is the Gauss-Tchebychev collocation (equation 3.5). The Gauss-Tchebychev collocation corresponds to the zeros of Tchebyshev polynomials and is optimal in the sense of Gauss quadrature.

$$x_i(k) = -1 + k\frac{2}{N-1} \tag{3.4}$$

$$x_i(k) = -\cos\frac{2k+1}{2N}\pi \tag{3.5}$$

Starting from $\mathscr{K}_D$ we intend to construct a new orthonormal basis by applying the Gram-Schmidt process. That implies that we need some product and norm for real bivariate functions defined on Ω. Taking into account the computational contingencies, given two real bivariate functions, F and G, their discrete extended inner product is defined by:

$$\langle F \,|G\rangle_w = \sum_{(u,v)\in D_1} F\left(\mathbf{x}_{(u,v)}\right) G\left(\mathbf{x}_{(u,v)}\right) w\left(\mathbf{x}_{(u,v)}\right) \tag{3.6}$$

with w a real positive function over Ω (Legendre, Tchebychev, Hermite, ...). Several weighting functions are shown in Table 3.1.

Then, the actual construction process of an orthonormal basis:

$$\mathrm{B}_{D_1,w} = \left\{ B_{d_1,d_2} \right\}_{(d_1,d_2)\in D_1} \tag{3.7}$$

Family	Ω	$w(x_1,x_2)$
Legendre	$[-1;1]^2$	1
Thecbychev 1	$[-1;1]^2$	$\frac{1}{\sqrt{(1-x_1)^2(1-x_2)^2}}$
Tchebycev 2	$[-1;1]^2$	$\sqrt{(1-x_1)^2(1-x_2)^2}$
Laguerre	$[0;\infty]^2$	$e^{-(x_1+x_2)}$
Hermite	$[-\infty;\infty]^2$	$e^{-\left(\frac{x_1^2+x_2^2}{2}\right)}$

Table 3.1 Some weighting functions w to obtain different families of polynomials

is a recurrence upon (d_1,d_2):

$$T_{d_1,d_2} = K_{d_1,d_2} - \sum_{(l_1,l_2)\prec(d_1,d_2)} \left\langle K_{d_1,d_2} \middle| B_{l_1,l_2} \right\rangle_w B_{l_1,l_2} \tag{3.8}$$

$$B_{d_1,d_2}(x) = \frac{T_{d_1,d_2}}{\left|T_{d_1,d_2}\right|_w} \tag{3.9}$$

where $\prec$ is the lexicographical order and $||_w$ the norm induced by $\langle|\rangle_w$.

Another method is to apply the three terms recurrence procedure (also called Stieljes process in the discrete case where the integral is calculated by elementary transposition):

$$\begin{cases} B_{-1,j}(x_1,x_2) = 0 \\ B_{i,-1}(x_1,x_2) = 0 \\ B_{0,0}(x_1,x_2) = 1 \\ B_{i+1,j}(x_1,x_2) = (x_1-\lambda_{i+1,j})B_{i,j}(x_1,x_2) - \mu_{i+1,j}B_{i-1,j}(x_1,x_2) \\ B_{i,j+1}(x_1,x_2) = (x_2-\lambda_{i,j+1})B_{i,j}(x_1,x_2) - \mu_{i,j+1}B_{i,j-1}(x_1,x_2) \end{cases} \tag{3.10}$$

where the coefficients λ and μ are given by:

$$\begin{array}{ll} \lambda_{i+1,j} = \frac{\left\langle x_1 B_{i,j} \middle| B_{i,j} \right\rangle}{\left\langle B_{i,j} \middle| B_{i,j} \right\rangle} & \lambda_{i,j+1} = \frac{\left\langle x_2 B_{i,j} \middle| B_{i,j} \right\rangle}{\left\langle B_{i,j} \middle| B_{i,j} \right\rangle} \\ \mu_{i+1,j} = \frac{\left\langle B_{i,j} \middle| B_{i,j} \right\rangle}{\left\langle B_{i-1,j} \middle| B_{i-1,j} \right\rangle} & \mu_{i,j+1} = \frac{\left\langle B_{i,j} \middle| B_{i,j} \right\rangle}{\left\langle B_{i,j-1} \middle| B_{i,j-1} \right\rangle} \end{array} \tag{3.11}$$

The resulting set of B polynomials verifies:

$$\left\langle B_{d_1,d_2} \middle| B_{l_1,l_2} \right\rangle_w = \begin{cases} 0 & \text{if } (d_1,d_2) \neq (l_1,l_2) \\ 1 & \text{if } (d_1,d_2) = (l_1,l_2) \end{cases} \tag{3.12}$$

B_{D_1}, w is effectively an orthonormal basis with respect to a weighting function w. A special case is the *complete basis* where D_1 represents exactly the set of pairs associated to Ω, that is

$$D_1 = [0;N_1] \times [0;N_2] \tag{3.13}$$

A complete basis, related to the discrete extended inner product (3.6) is the orthonormal basis whose domain is Ω defined by the family:

$$\left\{B_{d_1,d_2}(\mathbf{x})\right\}_{\substack{d_1=0..n_1 \\ d_2=0..n_2}} \tag{3.14}$$

The number of polynomials in the complete polynomial basis is given by $(n_1+1)\times(n_2+1)$. A tabular representation of such a basis is given Figure 3.1. The first $P_{0,0}$ polynomial is in the top left corner. An evolution along lines of a same column varies the degree of polynomials according to x_1. An evolution along columns of a same line varies the degree of polynomials according to x_2. The shape of Hermite 3 complete basis polynomials is presented in Figure 3.2.

$B_{N_1,N_2} =$

	$(x_2)^0$	$(x_2)^1$	$\dots$	$(x_2)^{(N_2-1)}$	$(x_2)^{N_2}$
$(x_1)^0$	$P_{0,0}$	$P_{0,1}$	$\dots$	P_{0,N_2-1}	P_{0,N_2}
$(x_1)^1$	$P_{1,0}$	$P_{1,1}$	$\dots$	P_{1,N_2-1}	P_{1,N_2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$(x_1)^{(N_1-1)}$	$P_{N_1-1,0}$	$P_{N_1-1,1}$	$\dots$	P_{N_1-1,N_2-1}	P_{N_1-1,N_2}
$(x_1)^{N_1}$	$P_{N_1,0}$	$P_{N_1,1}$	$\dots$	P_{N_1,N_2-1}	P_{N_1,N_2}

Figure 3.1 Tabular representation of a two-dimensional basis level $D_1 \times D_2$.

Properties of the complete basis

1. The complete basis allows to obtain, for a given set of data an interpolating function that is a first order polynomial (*ie* such as $\forall \mathbf{x}_{(u,v)} \in D, P_I\left(\mathbf{x}_{(u,v)}\right) = I\left(\mathbf{x}_{(u,v)}\right)$) if $D' = D$;
2. Furthermore the projections on $B_{i,j}$ can be considered as a multiscale finite differences operator $\partial_1^i \partial_2^j$.

3.2 Polynomial decomposition

If some of a texture characteristic parameters are obtained directly from the intensity levels of the pixels of the image, other important features can be known only through a filtering of

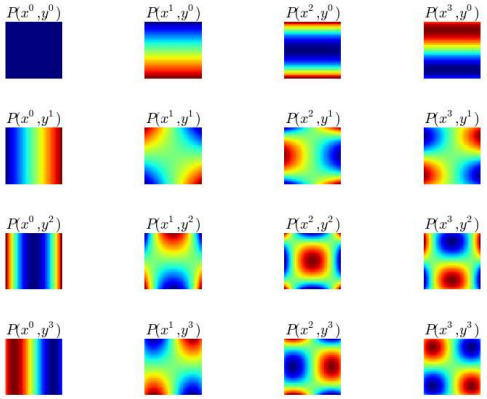


Figure 3.2 Polynomials of a Hermite 3 complete basis

the image. To setup an efficient strategy in image analysis, we need a joint spatial/frequency representation.

In this section, we show that real discrete orthonormal polynomials can be considered as a discrete multiscale decomposition, which allows us to represent texture information in a compact and accurate way.

We describe now the construction of the multiresolution piecewise polynomial decomposition.

Considering a function U defined on a domain Ω of size $n_1 \times n_2$ and a basis of size $h_1 \times h_2$, the decomposition process is expressed, at a step L, according to:

1. partition of the discrete domain Ω^L with a number of Δ sublattices , of sizes $h_1^L \times h_2^L$;
2. for each subinterval Δ, approximation of the corresponding restriction U^L in a complete basis constructed on Δ. The polynomials coefficients are defined as:

$$b_{i,j}(U^L) = \left\langle U^L \middle| B_{i,j} \right\rangle_w \tag{3.15}$$

3. the reordering or orthogonal polynomial coefficients b into $h_1^L \times h_2^L$ functions $U_{i,j}^{L+1}$, on domains of

$$\left[n_1^{L+1} \equiv \frac{n_1^L}{h_1^L} \right] \times \left[n_2^{L+1} \equiv \frac{n_2^L}{h_2^L} \right] \tag{3.16}$$

sizes to provide image subbands in a multiresolution decomposition-like structure.

This approach is equivalent to a filter bank: (i) the decomposition is obtained by filtering the image using the "polynomial" filters and (ii) the corresponding magnitude frequency responses of "polynomial" filters have additional low-pass and band-pass characteristics.

This technique provides a degree of flexibility which relates to the choice of resolution factors being potentially independent between different levels of decomposition. With respect to classic time-frequency representations, such as wavelets, polynomial basis decompositions do not necessarily use a dyadic partition and are therefore more adaptable. It should be noted that the coefficients of local polynomials can not be presented as a linear combination of upper level coefficients (as for classical wavelet transform).

To compute the polynomial coefficients we will use a regular grid subdivision that will be projected in the base. This technique provides a zero error approximation within acceptable computing times. This technique was successfully used in JPEG image compression format where the Discrete Cosine Transform is applied to blocks of 8×8 pixels.

Two examples of a first level decomposition on the same image are shown in Figure 3.3 with a decomposition using a 3×3 Chebychev complete basis (left) and a 5×4 Hermite complete basis(right).

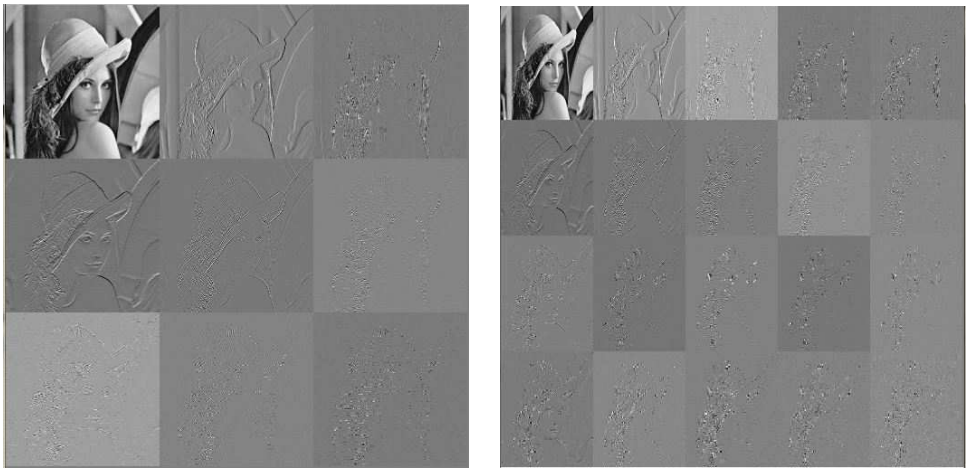


Figure 3.3 Two examples of a first level decomposition with Chebychev 3x3 and Hermite 5x4 complete basis

The computation of the polynomial projections $p_{i,j}(U^s) = \left\langle U^s \middle| B_{i,j} \right\rangle_w$ can be realized in form of convolution products with filters given by the values of the base polynomials measured at the collocation points.

For example using a Legendre complete basis of a 2×2 support, and a uniform collocation weighting function we obtain the next four filters:

$$f_{0,0} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \quad f_{0,1} = \begin{pmatrix} -0.5 & -0.5 \\ 0.5 & 0.5 \end{pmatrix}$$
$$f_{1,0} = \begin{pmatrix} -0.5 & 0.5 \\ -0.5 & 0.5 \end{pmatrix} \quad f_{1,1} = \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix} \tag{3.17}$$

Using this basis, we can observe on the left figure of Figure 3.4 a representation similar to Haar wavelets. The adaptability and therefore originality of the polynomial decomposition can be observed on the right images of Figure 3.4 and 3.3.

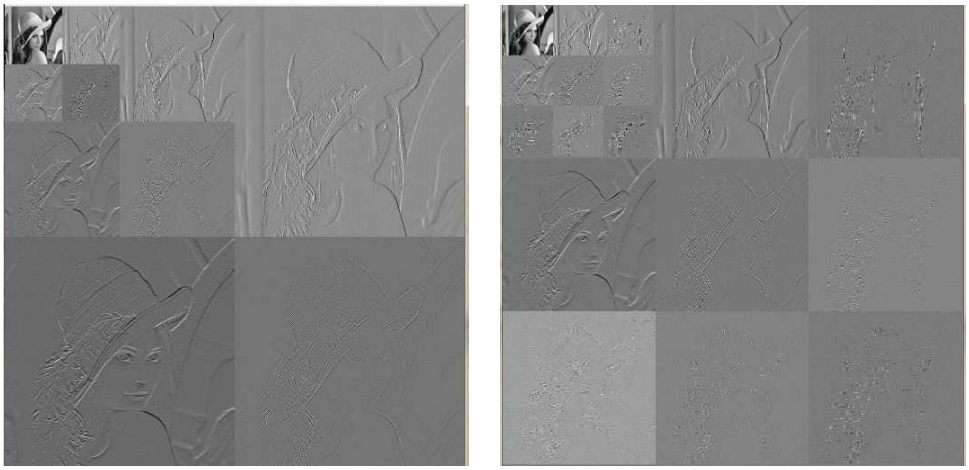


Figure 3.4 Example of third level 2×2 Legendre decomposition and second level 3×3 Tchebycev polynomial decomposition

3.3 Polynomial approximation

We have seen in the last section that by projecting an image in the complete basis we obtain a set of polynomial coefficients. We can reverse this operation and reconstruct the image using :

$$\tilde{U}(x_1,x_2) = \sum_{\substack{(d_1,d_2)\in[0;d] \\ d_1+d_2\leq d}} b_{d_1,d_2}(U)B_{d_1,d_2}(x_1,x_2) \tag{3.18}$$

where b_{d_1,d_2} are calculated using Equation 3.15.

During image reconstruction after after a multi-scale decomposition performed with complete bases, a perfect approximation is obtained by taking the entire set of calculated polynomial projection coefficients. Therefore, there is the possibility to use image approximations limiting the number of coefficients (and hence the number of polynomials) during the reconstruction phase. Different strategies can be used to select the polynomial coefficients:

- Brute force restriction to certain set of coefficients, for example those corresponding to polynomials whose degree is less than a threshold

- Restriction based on "energies", for example by using the basis normality to compare energy quantities that are associated to each domain coefficients keeping only a fixed number of coefficients or those that satisfy a certain condition (energy)

We present, in figure 3.5 the perfect approximation of an image using complete bases with a grid of $2\times$ 2 pixels. We therefore use a base containing 4 polynomials which are $P_{0,0}, P_{0,1}, P_{1,0}, P_{1,1}$. The test image is presented in figure 3.5(a) and in 3.5 (b) the result of the approximation. It can be noted that the approximation error is equal to zero. This result is identical regardless of the grid used (and thus the number of polynomials).

(a) (b)

Figure 3.5 Approximation of Lena image (512 $\times$ 512 pixels) using complete basis on a regular 2×2 grid.(a) Original Lena image; (b) Image reconstruction

3.3.1 Experimental results on image approximations

We decide to compare the polynomial approximation with Haar wavelet transform, used by Wolstenhome ant Taylor [WT99] to build a wavelet appearance model, CDF (Cohen-Daubechies-Faveau) 9-7 wavelets used in JPEG2000 codec for lossy compression and integrated by [SFC04] in the AAM framework and SVD (singular value decomposition), a compression method having the property of energy compaction and the ability to adapt to the local statistical variations of an image. SVD is closely related to PCA, the method used in AAM to describe variations in shape and texture.

Haar wavelets The Haar wavelet is one of the simplest and oldest wavelets. It's mother wavelet function $\psi(t)$ and its scaling function $\phi(t)$ can be described as:

$$\psi(t) = \begin{cases} 1, & 0 \leq t \leq \frac{1}{2} \\ -1, & \frac{1}{2} \leq t \leq 1 \\ 0, & \text{otherwise} \end{cases} \qquad \phi(t) = \begin{cases} 1, & 0 \leq t \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Any filter can be approximated uniformly by linear combinations of Haar wavelets.

CDF 9/7 wavelets The CDF 9/7 wavelet is an especially effective biorthogonal wavelet, used by the FBI for fingerprint compression and selected for the JPEG2000 standard. The reason CDF 9/7 wavelets are used in practice is because they were designed to come very close to being energy preserving. The low pass filters associated with the CDF 9/7 wavelets have 9 coefficients in the analysis, and 7 coefficients to synthesize. This wavelet has a great number of null moments for a relatively short support. The scaling functions and wavelets of the CDF 9/7 wavelets are presented in the Figure 3.6.

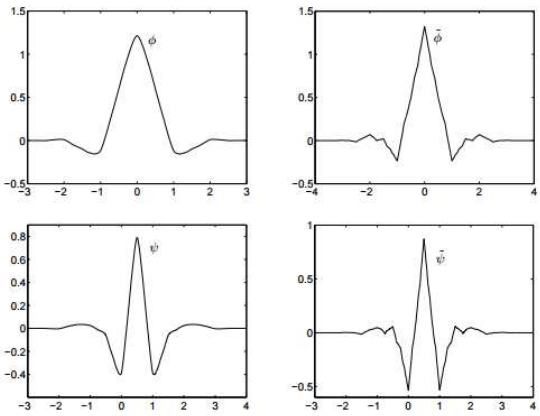


Figure 3.6 CDF 9/7 scaling functions and wavelets

SVD Singular value decomposition is a linear algebra method for identifying and ordering the dimensions along which data points exhibit the most variation. The SVD values represent the energy of an image. Indeed, the total energy of an image I can be represented by :

$$\|I\| = trace\left[I^T \times I\right] = \sum_{i=1}^{m}\sum_{j=1}^{n} I^2(i,j) = \sum_{i=1}^{n} \sigma_i^2$$

Image compression of a grayscale image is conducted by forcing the low singular values to zero. Keeping the $k, (k \leq n)$ first singular values an image I can be approximated using:

$$\|I_k\| = trace\left[I_k^T \times I\right] = \sum_{i=1}^{k} \sigma_i^2$$

We conduct our tests on an image of the sixth person of the database MUG [APD10b] performing the neutral expression and Lena image.

The quality of images is evaluated using PSNR("Peak Signal to Noise Ratio") measure:

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{3.19}$$

where MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is equal to 255.

In our tests we construct a Hermite 16×16 basis. The decomposition obtained after projection on this basis is similar to a 4-levels decomposition using CDF 9/7 and Haar wavelets.

First we show the outcome of the comparison of the reconstruction using complete basis, Haar and CDF 9/7 wavelets using a brute force restriction on the coefficients, and keeping the ones that correspond to a multi-resolution level. In this case, if the coefficients after a transform are denoted by :

$$coeffs = [a\ u_1 \ldots u_n] \quad \text{where} \quad u_n = [h_n v_n d_n]$$

where a, h, v, d denote approximation, horizontal, vertical and diagonal detail coefficients respectively. After truncation, the selected coefficients will correspond to : $coeffs = [a\ u_1 \ldots u_{restriction}]$

The comparison results are presented in Figures 3.7, 3.8 and 3.9. It can be observed that using this type of restriction, our method slightly outperforms the Haar wavelet transform , and is similar to the CDF 9/7 wavelet transform. Using the low level coefficients the best results are obtained with the CDF 9/7 wavelet, Haar wavelets and complete basis having the same PSNR. By adding detail coefficients, complete bases outperform Haar wavelets, and for the face image, even the CDF 9/7 wavelet transform.

In Figure 3.10, we present the results based on the energy selection of the coefficients. After the decomposition, we keep the best coefficients that correspond to a fixed amount of energy stored in each block. We do not remove any coefficients from the level zero decomposition (corresponding to the low frequency values of the signal). After this step we compute the PSNR between the original image and the reconstructed one. In the results presented

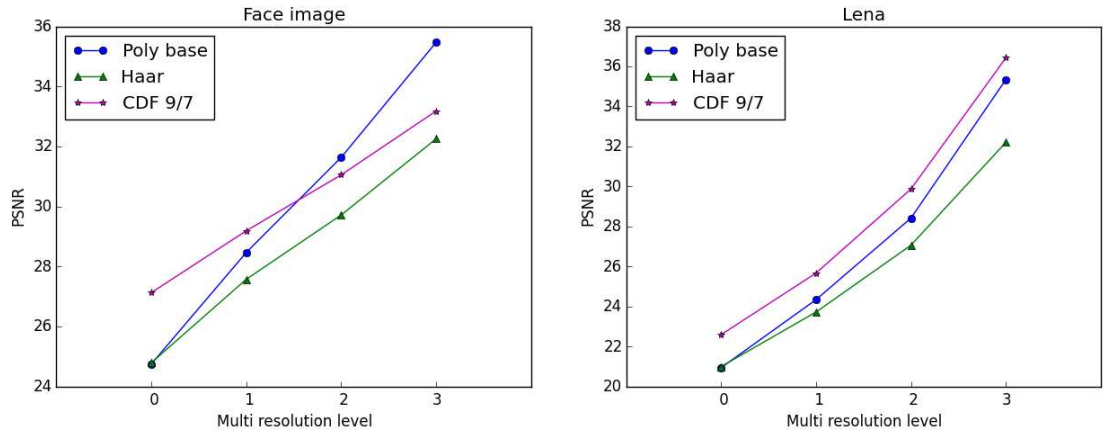


Figure 3.7 PSNR evolution using the brute force restrictions on coefficients for Lena and MUG face image

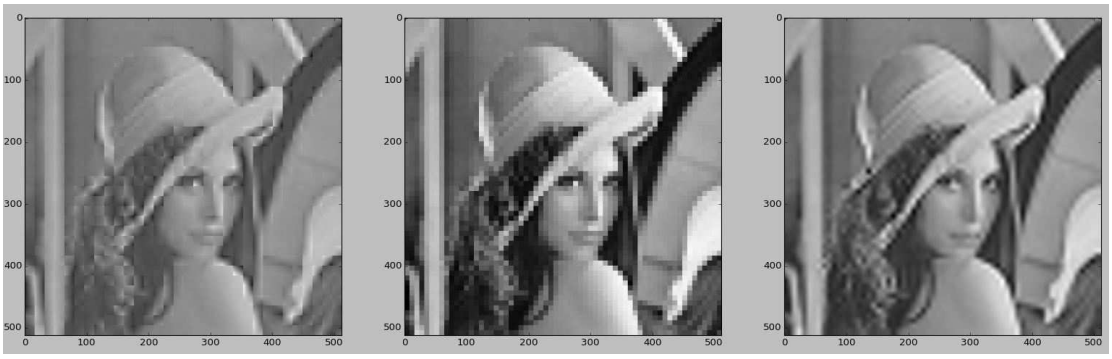


Figure 3.8 Reconstruction using 0 and 1 level coefficients after a 4 level decomposition. From left to right: complete basis, Haar an CDF 9/7 wavelets

below, we use a SVD on the entire image, the approximated image being reconstructed keeping the vectors that correspond to a fixed amount of energy.

It can be seen that using the energy based restriction, polynomial approximation outperforms the other methods. The SVD process gives the worse results. This is obtained because when using other transform we can select more precisely the coefficients to keep, while using SVD we have to truncate the obtained matrix.

The results keeping 1% and 15% of the coefficients are presented in Figures 3.11 and 3.12. Following the results presented above, and since polynomial base decompositions do not necessarily use a dyadic partition like wavelets do and are therefore more adaptable, we can conclude that image approximations with complete basis are a good tool for image approximation.

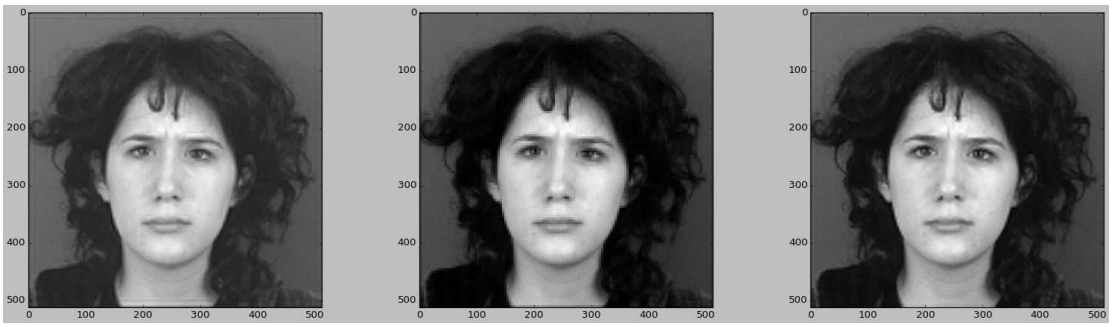


Figure 3.9 Reconstruction using 0-2 levels coefficients after a 4 level decomposition. From left to right: complete basis, Haar an CDF 9/7 wavelets

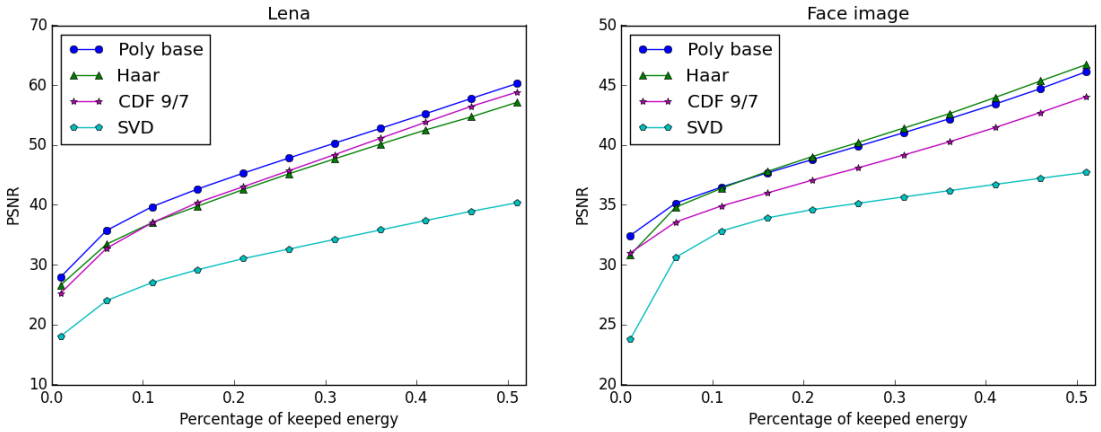


Figure 3.10 PSNR evolution using a fixed amount of energy coefficients for Lena and MUG face image

3.3.2 Discussion

In this chapter we have presented the complete bases that are the basic tool used during the thesis. We showed how real discrete orthonormal polynomials can be considered as a discrete multiscale decomposition allowing the representation of texture information in a compact and accurate way and presented an application of the polynomial approximation for image compression.

By comparing our method to 3 different approaches - Singular Value decomposition, Haar and CDF 9/7 wavelets we have demonstrated the capacity of polynomial coefficients to represent an image in a sparse, compact and in a flexible manner (due to sizes of the basis).

Figure 3.11 Reconstruction using 1 percent of the coefficients. From left to right: SVD, complete basis, Haar an CDF 9/7 wavelets

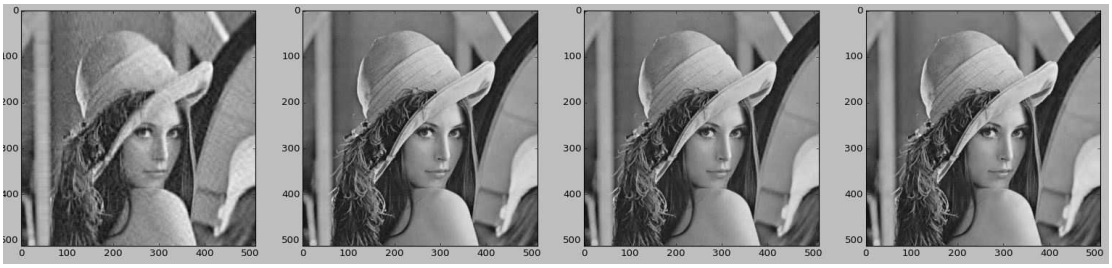


Figure 3.12 Reconstruction using 15 percents of the coefficients.From left to right: SVD, complete basis, Haar an CDF 9/7 wavelets

Polynomial Active Appearance Models

Chapter 4

Face texture analysis with polynomials

In this part, we propose a new polynomial texture representation method for deformable models. While many texture representations have been proposed over the years to improve the accuracy and reliability of computer vision applications such as object tracking or image alignment, most descriptors are usually unable to both provide precise multi-scale and multi-orientation analysis and handle the redundancy problem effectively.

Because model fitting parameters within AAMs are estimated by minimizing the sum of squared differences of texture values between observations and approximations, accuracy and robustness will rely heavily on choices regarding texture representation. While original AAMs use raw image intensities, and despite the advantages of PCA-based global structure analysis, the AAM fitting is still an open issue (*e.g.* it is very sensitive to illumination and pose changes), as useful information on local variations, such as their scale or orientation, is either neglected or not extracted well enough by PCA.

We will demonstrate the integration of coefficients obtained from polynomial projections both into the fitting procedure of discriminative and generative approaches, and their ability to improve robustness against pose and facial expression changes.

4.1 More on Active Appearance Models

We presented briefly the AAM algorithm in the last chapter. We will see now this algorithm more thoroughly. An Active Appearance Model (AAM) is a statistical model of shape and texture (appearance) learned from a training set comprising different views of an object, combined with a deformation algorithm to match new images. It was first introduced by Cootes et al. in 1998 [ETC98] and has been widely used in face tracking and medical imaging applications ever since. There are two basic components in alignment using an AAM: **data modeling** and **model fitting**.

4.1.1 Data modelling

Given a set of training images, data modeling is the procedure of training the AAM, which is essentially two distinct linear subspaces modeling object shape and appearance respectively. The variations of geometry and texture are modeled using linear modes of deformation, computed from a Principal Component Analysis (PCA) on the training data. Advantageously, separate models are built for the object shape (or geometry) and its appearance (or texture). For face images, the shape is defined as the set of v 2D coordinates of the landmarks used for model building $\mathbf{s} = [x_1, y_1, x_2, y_2, ..., x_v, y_v]$. This set of landmarks can be triangulated, for instance using Delaunay triangulation, to provide a mesh (see Figure 4.1). As illustrated on Figure 4.1, taken from [MB04], the shape model learnt from the annotated training images consists of a base shape $\mathbf{s_0}$ and a set of linear modes of deformation $\mathbf{s_i}$ of this base shape. Thus, the model can represent any linear combination of the basis shapes, of the form

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} r_i \mathbf{s}_i \tag{4.1}$$

where $\mathbf{r} = [r_1, r_2, ..., r_n]$ are the shape parameters. By design, the first four shape basis vectors can represent global scale, rotation and translation. Together with other basis vectors, a mapping function from the model coordinate system to the coordinates in the image observation is defined as $\mathbf{W}(\mathbf{x}; \mathbf{r})$, where $\mathbf{x}$ is a pixel coordinate defined by the mean shape s_0. Any point $\mathbf{x}$ in s_0 must lie within one of the triangles in the mesh associated to s_0. The images by $\mathbf{W}$ of the vertices of the triangles are known from the expression of $\mathbf{s}$. The restriction of $\mathbf{W}$ to this triangle is defined to be the affine transform computed from the images of its vertices.

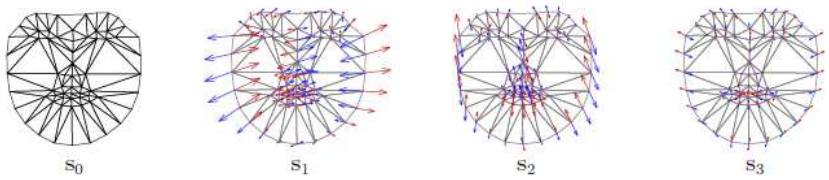


Figure 4.1 Deformable model of shape in an AAM. Taken, from [MB04]

The annotation for each training face image provides a training mesh. In order to compute the shape model, the training shapes defined by the set of annotated landmarks are first registered to a common position, scale and orientation using Global Procrustes Analysis. $\mathbf{s}_0$ is defined as the mean of these registered shapes, the main modes of deformation $\mathbf{s_i}$ are computed using a principal component analysis.

The appearance in an AAM consists of the set of pixel intensities within the convex hull of the base shape $\mathbf{s}_0$. Similarly to the shape, and as illustrated on Figure 4.2, the appearance

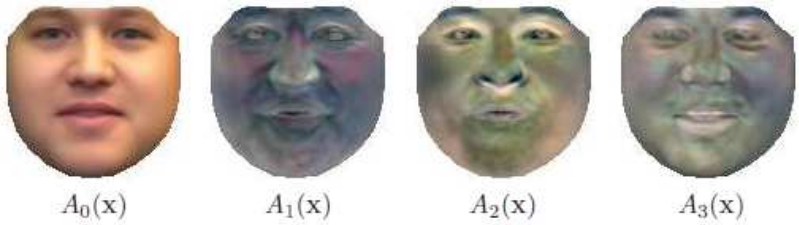


Figure 4.2 Deformable model of appearance in an AAM. Taken, from [MB04]

model is defined by a base appearance $\mathbf{A_0}$ and a set of linear modes of deformation $\mathbf{A_i}$ of this appearance. Thus, the model can represent any appearance of the form :

$$\mathbf{A}(\mathbf{x},\lambda) = \mathbf{A_0}(\mathbf{x}) + \sum_{\mathbf{i=1}}^{\mathbf{m}} \lambda_\mathbf{i}\mathbf{A_i}(\mathbf{x}) \tag{4.2}$$

where $\mathbf{A_0}$ is the mean appearance, $\mathbf{A_i}$ is the i^{th} appearance basis and $\lambda = [\lambda_\mathbf{1}, \lambda_\mathbf{2}, ..., \lambda_\mathbf{m}]$ are the appearance parameters. It is worth emphasizing at this point that the appearance part of the model is defined within the base shape $\mathbf{s_0}$, and therefore independent of the shape.

The AAM model instance with shape parameters $\mathbf{r}$ and appearance parameters λ is created by warping the appearance $\mathbf{A}$ from the base mesh $\mathbf{s_0}$ to the model shape $\mathbf{s}$ using a piecewise affine warp denoted $\mathbf{W}(\mathbf{x};\mathbf{r})$.

The sequence of operations needed to generate an instance of the model from a set of parameters (r_i, λ_i) are as follows:

1. Generate the shape as $\mathbf{s} = \mathbf{s}_0 + \sum \lambda_i \mathbf{s}_i$
2. Generate the appearance within $\mathbf{s_0}$ as $\mathbf{A} = \mathbf{A_0} + \sum \lambda_\mathbf{i}\mathbf{A_i}$
3. Compute the warp $\mathbf{W}(\mathbf{x};\mathbf{r_i})$ that maps $\mathbf{s_0}$ to $\mathbf{s}$
4. Warp the appearance $\mathbf{A}$ from the base shape $\mathbf{s_0}$ to the model shape $\mathbf{s}$ using $\mathbf{W}$.

4.1.2 Model fitting

Model fitting refers to estimating the best deformation parameters that allow to fit a trained model to an input image.

In order to localize landmarks on new images the model learnt from the training set is fitted to the images. In the case of AAM, fitting stands for computing the shape and appearance model parameters that best "explain" the input image. More precisely, a hypothesized choice of model parameters defines a model instance, whose geometry delimits the face region

within the input image, and whose appearance provides a reconstructed face texture in $\mathbf{s_0}$. The optimal parameters are those that minimize the squared error between the $\mathbf{A}$ and the warping onto $\mathbf{s_0}$ of the object region in the input image.

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{x})]^2 \tag{4.3}$$

There are two major research lines for modeling this function.

The first one is using a standard gradient descent algorithm, iteratively updating the current estimate, until convergence. This line of research is referred as **generative fitting**. Because straightforward implementations of gradient descent are computationally expensive and therefore very slow, most approaches to generative AAM fitting either assume some parts of the model are fixed, or reformulate the problem so that they become fixed. The approach taken in [MB04] is to start from a coarse initialization of the mesh and apply an iterative optimization scheme to minimize, over the set of model parameters $(\mathbf{r_i}, \lambda_\mathbf{i})$, the reconstruction error between the appearance model instance (defined by the $\lambda_\mathbf{i}$) and the observed set of pixels within the shape model instance, warped back to the base shape $\mathbf{s_0}$ according to the values of the shape parameters ($\mathbf{r}$). In their project-out inverse compositional method the key idea is that the role of the appearance template and the input image is switched when computing parameter increment $\Delta\mathbf{r}$ and because it decouples shape from appearance by projecting out appearance variation. This enables several time-consuming steps of parameter estimation to be pre-computed and performed outside of the iteration loop and therefore their approach is often considered the standard choice for fitting person-specific AAMs.

The second line of research for fitting AAMs is through learning the error function via regression. This type of technique is fast but approximate and is referred as **discriminative fitting**. In order to take into account the dependency to texture differences, without having to recompute it for every new image, the original AAM formulation in [CET01b] proposes to learn the relationship between the spatial pattern in the error image and the way the parameters should be changed, in other words, to learn a gradient matrix from a set of training examples. This technique is fast but approximate, because the error function is linear and independent of the current model parameters. A notable improvement is the work of [SG07] in which a nonlinear regressor is learned via boosting. Other discriminative methods for fitting AAMs have been proposed in [Liu07] and [WLD08].

4.2 Texture representation in discriminative AAMs

In this section, we propose two different schemes to replace the original AAM texture representation model by approximating image structures with polynomial projections on an orthonormal basis.

4.2.1 Background on texture representations

In recent years, several alternative texture representation methods have been proposed to improve the accuracy and reliability of the matching. As an example, Cootes and Taylor [CT01] use a combination of gradient, absolute edge and a cornerness. Kittipanya-Ngam and Cootes [KnC06] show that representations of the structure of the image in a region may improve the fitting process of AAMs, and a half-wave rectified gradient is presented. According to Stegmann and Larsen [SL03], a mix of features (*e.g.* intensities and contours) gives better results than any individual representation, while Su and Tao [STLG09] propose a representation that combines Gabor wavelets with Local Binary Patterns (LBP).

While statistical learning methods such as Principal Component Analysis (PCA) perform very well in global structure relationship studies, it has also been demonstrated that computer vision models can greatly benefit from local analysis and hierarchical representations of image structures obtained after image convolution with a set of filter banks. This type of processing is motivated by two widely accepted assumptions about human vision: (i) human vision is mostly sensitive to scene reflectance and mostly insensitive to the illumination conditions, and (ii) human vision responds to local changes in contrast rather than to global brightness levels. A natural way to encode local contrast is through the employment of a bank of filters that encode local intensity differences at different orientations and scales. As an example, the use of wavelets can enhance the accuracy and robustness of AAM models [HFT+03, SFC04, DAVM04, STLG09].

Gabor wavelets have also been used successfully by Davoine et al. within AAMs [DAVM04] to build a hierarchical model based on a set of filter responses. Because they provide an efficient framework for multi-scale and multi-orientation structural analysis, they are usually robust against illumination and pose changes, and are widely used in applications such as facial expression recognition. However, their non-orthogonal decomposition makes them unable to handle the redundancy problem effectively. For this reason, we propose to study and use a more compact and adaptive representation for deformable models, namely the 2D polynomial transform. Later in this section, we will show how we intend to use it.

4.2.2 Proposed approach

Our motivation to use a texture polynomial representation in the AAM framework is that orthogonal polynomials have some properties related to the human visual system [Bla74], including a multi-scale / multi-resolution representation of the information. We search a deformable model that could capture a structure with an insignificant number of parameters, and therefore we propose to integrate a polynomial representation for its coding qualities, namely because of the compact representation of the information in the polynomial model.

The proposed method is built on two major components: the adaptation of AAM to incorporate texture features and the generation of texture features using the polynomial bases.

In equation (4.2) we replace the texture model $\mathbf{A}$ by $\mathbf{A_P}$, a vector of approximation coefficients obtained through polynomial projections on aligned textures. Two different modes are available for the computation of coefficients of $\mathbf{A_P}$:

- calculated on texture patches sampled around key landmarks (PAAM)
- retrieved from a multi-resolution polynomial decomposition of the full aligned texture. (FT-PAAM)

In the first case (PAAM) the texture sampling method is different from the one used for appearance calculation described previously. A training patch is sampled around each landmark, and projected on a polynomial basis. The size and the collocation function used for basis generation will be discussed later. The texture patches obtained by polynomial projections from a given training image are then concatenated to form a single grey value vector. Then the set of grey scale training vectors and normalised shape co-ordinates are used to construct linear models, as seen in the model building step of the AAM Algorithm. This type of texture representation will reduce the model basis dimensions to avoid over fitting.

As for the FT-PAAM approach, depending of the size of the complete base, we will obtain a defined number of sublattices, and the texture will be considered as the matrix of concatenated polynomial coefficients calculated on the subdomains. As the projections on polynomials of a complete base can be considered as a multi-scale finite difference related to differentials $\partial_1^i \partial_2^j$ the coefficients obtained after the projection in the polynomial base could be used directly for computations requiring gradients.

4.2.3 Experimental results

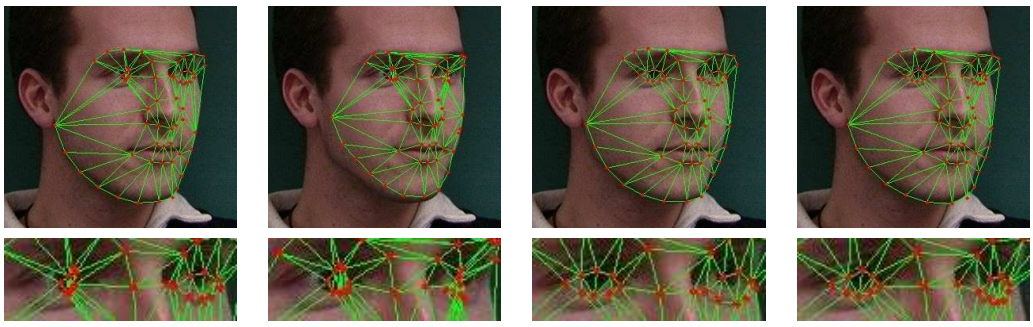
For these experiments, 20 different AAM models have been trained: 5 on 30 images out of 5 randomly selected subjects from the IMM database, 5 on 30 frontal images of 5 subjects

	MultiPie	**MUG**	**IMM**	**CK**
Cootes et al.	2.248 ± 0.740	1.735 ± 0.413	1.972 ± 0.512	1.611 ± 0.754
ASM	2.461 ± 0.655	1.738 ± 0.422	3.356 ± 2.142	1.921 ± 0.649
ICIA	1.958 ± 1.080	1.946 ± 1.041	1.984 ± 0.977	$\mathbf{1.411} \pm 1.091$
PAAM (ours)	2.261 ± 0.709	1.838 ± 0.374	$\mathbf{1.904} \pm 0.504$	1.630 ± 0.788
FT-PAAM (ours)	$\mathbf{1.887} \pm 0.604$	$\mathbf{1.480} \pm 0.377$	1.925 ± 0.822	1.601 ± 1.260

Table 4.1 Mean Error pixel/landmark

with various emotions from the MUG database, 5 on 30 images of 30 different subjects performing various facial expressions from the Cohn Kanade database and the other 5 on 30 images of 5 subjects with different facial expressions from the MultiPie database.

The training and fitting databases that we use are very challenging as we have multi-user /multi expression and multi-user/multi pose variations. For each database, the 5 computed models are the original Cootes et al. ASM [CTCG95] and AAM model [CET01a], Matthews-Baker's Inverse Compositional Image Alignment (ICIA) model [MB04], our polynomial projection-based model (PAAM) and the first level polynomial decomposition coefficients of the full aligned texture (FT-PAAM).

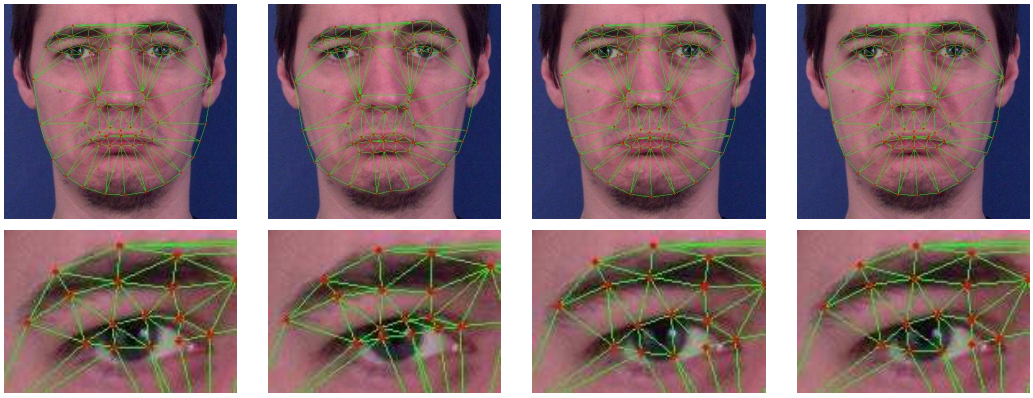


Cootes et al. 1.94031 | ICIA 2.13562 | PAAM (ours) 1.58384 | FT-PAAM(ours) 1.58442

Figure 4.3 Face matching example on IMM database

The polynomial approximation coefficients used for the PAAM representation were obtained via projections on a 15×15 complete Hermite basis - a reasonable size for modeling local changes in texture. Approximation coefficients are calculated on texture patches around key landmarks, and deformation parameters are estimated through a Cootes-like Taylor approximation. As for FT-PAAM we use a 3x3 Laguerre complete basis sufficient for gradient computations. For both approaches we used Chebychev collocation method.

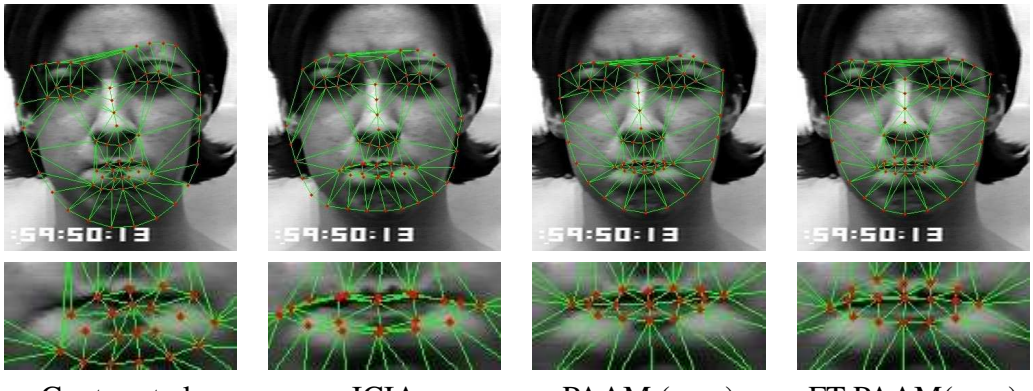
Results were calculated when the initialized shape was randomly perturbed from ground-truth. Table 4.1 shows the mean error and the standard deviation computed with the five different methods. As can be observed, our two proposed methods give very accurate results.



Cootes et al. 1.4142 | ICIA 1.51793 | PAAM (ours) 1.11305 | FT-PAAM(ours) 1.20436

Figure 4.4 Face matching example on MUG database

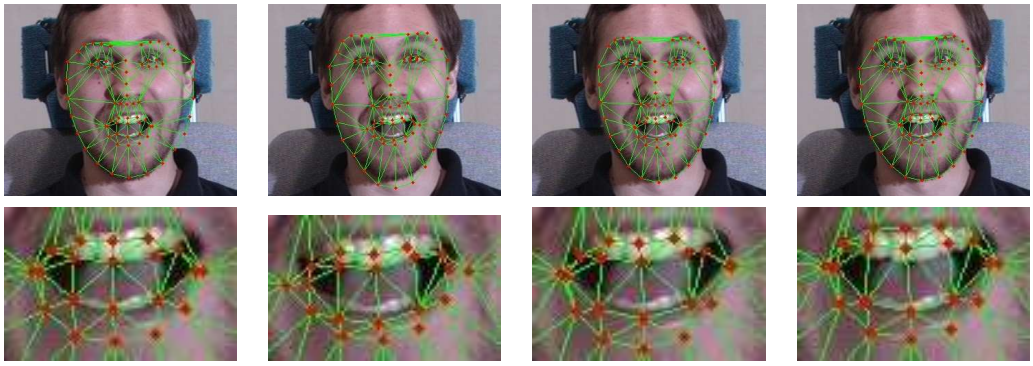
A noticeable difference between the results obtained with our two methods is that the PAAM scheme performs better when there is a pose variation. During tests we did notice significant improvements of our method over both Cootes et al. and ICIA methods on profile poses (see Fig. 4.3, the eyes location). In this approach, the AAM model is built on texture patches around the keypoints , so it can give more accurate results than the one calculated on the entire set of pixels of the AAM model, in particular in the case of the variation of expression or facial pose. As we use spatially localized texture models around the points of interest, our method must provide strength to the local modifications of texture.



Cootes et al. 2.83959 | ICIA 2.88878 | PAAM (ours) 1.42299 | FT-PAAM(ours) 1.49749

Figure 4.5 Face matching example on Cohn Kanade database

As can be seen in Fig.4.5 and Fig. 4.6 presenting the matching results on a subject of the CohnKanade and MultiPie database, the FT-PAAM locates the landmarks more accurately than do the other methods, especially for the points on the chin, that are quite difficult to fit and that are usually not considered in error metric calculations. This is explained by the fact that we have a hierarchical representation of the information when processing texture coefficients via polynomial projections.



Cootes et al.	ICIA	PAAM (ours)	FT-PAAM(ours)
1.57715	1.50364	1.30466	1.22499

Figure 4.6 Face matching example on CMU MultiPie database

4.3 Polynomial compressed appearance models

Active Appearance Models establish dense correspondences by modeling the variations between shape and pixels intensities and is an generative model that is able to synthesize a very close approximation to any image of the target object. Using raw image intensities to model each pixel of an object and despite the advantages of PCA-based analysis, this problem is computationally expensive for high resolution 2D images considering the computational requirements and the large data storage. Therefore, to reduce the computational time we propose to use the approximation coefficients of polynomial projections in place of the raw intensities. This operation will introduce an additional time requirement by cause of transformation of the image into a new representation. However [LSD+07] showed that the computational burden of model fitting can be considerably reduced if the transformation leads to sparse data.

In this part we to extend the work of Wolstenhome and Stegmann [SFC04] applied to face alignment where wavelet coefficient subsets were modelled rather than pixel intensities. Yet, deviating from their approach we will include the approximation coefficients in the regression framework.

The proposed method is built on two major components: the adaptation of AAM to incorporate texture features and therefore the generation of this features using the polynomial basis and the regression algorithm used for model fitting.

4.3.1 Compressed polynomial texture model

We have seen that the polynomial representation is able to encode the local texture information of an object by projections on the different polynomials of the basis. As a result, the polynomial representation is capable of characterizing the statistical properties of object appearance in multiple scales and directions (by adding an angle in the basis generation, similar to gabor wavelets). This section introduces a notation for the polynomial compression and describes how it can be integrated in the AAM framework .

First let a n-level (the overall degree of the polynomials in the basis) polynomial transform be denoted by :

$$P(\mathbf{t}) = \mathbf{p}_{i,j}(\mathbf{t}) = \hat{\mathbf{p}} = \left[\hat{\mathbf{B}}_{0,0}^T \hat{\mathbf{B}}_{0,1}^T \dots \hat{\mathbf{B}}_{d_1,d_2}^T\right]^T \tag{4.4}$$

where $\hat{\mathbf{B}}_{i,j}$ denote the polynomial coefficients of the texture $\mathbf{t}$ projected in the basis. Compression is now obtained by a truncation of the polynomial coefficients:

$$C(\hat{\mathbf{p}}) = C\mathbf{p} = \mathbf{p} = \left[\mathbf{B}_{0,0}^T \mathbf{B}_{0,1}^T \dots \mathbf{B}_{d_1,d_2}^T\right]^T \tag{4.5}$$

where C is a modified identity matrix, with rows corresponding to truncated coefficients removed. Notice that for simplicity reasons we do not use a special notation for compressed coefficients. However we used the $\hat{\mathbf{p}}$ symbol for the uncompressed transform.

The appearance model is built on the truncated polynomial coefficients constituting the texture. The texture PCA on polynomial coefficients is given by :

$$\mathbf{p} = \overline{\mathbf{p}} + \sum_{i=1}^{m} \lambda_p \mathbf{b}_p \Leftrightarrow \begin{bmatrix} \mathbf{B}_{0,0} \\ \mathbf{B}_{0,1} \\ \vdots \\ \mathbf{B}_{d_1,d_2} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{B}_{0,0}} \\ \overline{\mathbf{B}_{0,1}} \\ \vdots \\ \overline{\mathbf{B}_{d_1,d_2}} \end{bmatrix} + \sum_{i=1}^{m} \begin{bmatrix} \lambda_{B_{0,0}} \\ \lambda_{B_{0,1}} \\ \vdots \\ \lambda_{B_{d_1,d_2}} \end{bmatrix} \mathbf{b}_p \tag{4.6}$$

where $\lambda_{\mathbf{p}}$ is the eigenvectors of the polynomial coefficient covariance matrix. Rearranging this into low and high frequency terms we get:

$$\mathbf{B}_{0,0} = \overline{\mathbf{B}_{0,0}} + \lambda_{B_{0,0}} \mathbf{b}_p \tag{4.7}$$

the coarse approximation, and

$$\{\mathbf{B}_{d_1,d_2} = \overline{\mathbf{B}_{d_1,d_2}} + \lambda_{B_{d_1,d_2}}\mathbf{b}_p\}_{\substack{d_1=1..n_1 \\ d_2=1..n_2}} \tag{4.8}$$

the details. Hence the texture is multi-scale and can be used for analysis/synthesis at any given scale (provided by the polynomial degree). Truncating a substantial number of polynomial coefficients and compared to the multi-scale AAM this method can give a major reduction in storage requirements.

4.3.2 Iterative regression model using polynomial coefficients

In this section we introduce an algorithm for the training and evaluation procedures using a cascaded polynomial regression model that is inspired by Nonlinear Discriminative Fitting [SG07]. Regression-based methods directly learn a mapping function from facial image appearance to facial feature points.

Saragih and Goecke used a nonlinear update model for AAM fitting that uses multimodal weak learners, based on Haar-like features, which allow efficient online evaluation using the integral image. To avoid overlearning, the boosting procedure is embedded into an iterative framework with an intermediate resampling step. This process affords well regularised update models through limiting the ensemble size and indirectly increasing the sample size.

In this section we present the CDAAM, a framework for incorporating polynomial compression into a regression framework, using global shape and combined parameters of shapes and appearance. Polynomial approximation coefficients are used to compress the data; then PCA is used to reduce the dimensionality of the shape and appearance, and those projections are concatenated, similar to traditional AAM's.

In our approach, in addition to shape and appearance parameters we use a 2D similarity transformation which parameters q_p contains the rotation, translation, and scale parameters. Given an input pose S_0 we first estimate the shape parameters ie the global shape transform q_p and the combined shape and appearance parameters c_p. Then we train a regressor $R = (R_1;..;R_T)$, such that, given an input pose S_0 and its parameters q_{p0} and c_{p0} , the pose parameters at iteration t are evaluated by computing: $(q_{p,t}, c_{p,t}) = (q_{p,t-1}, c_{p,t-1}) + (\partial q_{p,t}, \partial c_{p,t})$ where $(\partial g_{q,t}, \partial c_{p,t})$ are computed via Lasso Regression.

Training

The training procedure for a CDAAM (compressed polynomial regression AAM) is shown in Alg. 3.

Algorithm 3 Training of cascaded polynomial regression

Input: N images with landmarks annotations

1: Build a statistical model of joined shape and compressed texture using polynomial projections
2: **for** $t = 1$ to T **do**
3: $\left\{\delta q_{p,t}, \delta c_{p,t}\right\}_{j=1}^{S}$ Sample perturbations
4: $\mathbf{f}_i = \mathbf{p}\left(I \circ \mathscr{W}(q_{p,t}, c_{p,t})\right)$ // compute compressed features using last fitted shape, where $\mathscr{W}$ is a warping function
5: **if** t>1 **then**
6: Estimate $(\delta q_{p,t}, \delta c_{p,t})$ using R_{t-1} and add them to the current parameters
7: **end if**
8: $(\tilde{q}_{p,t}, \tilde{c}_{p,t}) = (q_{p,t}, c_{p,t}) - (\delta q_{p,t}, \delta c_{p,t})$
9: $R_t = \arg\min_R \sum_i d(R(\mathbf{f}_i), (\tilde{q}_{p,t}, \tilde{c}_{p,t}))$
10: **end for**
11: Output $R = (R^1, ..., R^T)$

Each component regressor R_t is trained to attempt to minimize the difference between the true pose parameters and the pose parameters computed previously. In each iteration t we begin to generate small displacements $(\delta q_{p,t}, \delta c_{p,t})$ for each parameter from the known optimal value and learn a regressor R_t such that $(\partial q^t_{p,t}, \partial c^t_{p,t}) = R_t(\delta q_{p,t}, \delta c_{p,t})$ minimizes the following loss:

$$\min \sum_{i=1}^{N} |(q^T_{p,t}, c^T_{p,t}) - (q^0_{p,t}, c^0_{p,t})| \tag{4.9}$$

where $(q^T_{p,t}, c^T_{p,t})$ are the current shape parameters and $(q^0_{p,t}, c^0_{p,t})$ the true pose parameters.

In this work we rely on lasso regression, described below.

Lasso regression

The Lasso is a linear model that estimates sparse coefficients and in which the target value is expected to be a linear combination of the input variables. Given a set of input measurements $x_1, x_2 ... x_p$ and an outcome measurement y, the predicted value $\hat{y}$ is estimated via:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_p x_p \tag{4.10}$$

We decided to use it due to its tendency to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent. Given an outcome vector y, a matrix $\mathbf{X}$ of predictor variables, and a tuning parameter $\lambda \geq 0$, the lasso estimate can be defined as:

$$\arg\min_{\omega} \frac{1}{2} \|y - \mathbf{X}\omega\|_2^2 + \lambda \|\omega\|_1 \tag{4.11}$$

The lasso estimate thus solves the minimization of the least-squares penalty with $\lambda||w||_1$ added, where λ is a constant and $||w||_1$ is the ℓ_1-norm of the parameter vector and a coordinate descent as the algorithm to fit the coefficients.

Fitting

After the training step, given an input shape s_0, we project it to parameters and the regressor $R(S_0;I)$ is evaluated by computing: $(\delta g_{p,t}, \delta c_{p,t}) = R^t(x)$ from $t = 1...T$ and finally outputting S^T. (see Algorithm 4)

Algorithm 4 Evaluation of cascaded polynomial regression

Input: Image I, initial shape S^0 and the set of regressors $R = (R^1, ..., R^T)$

1: Project shape to compressed parameters
2: **for** $t = 1$ to T **do**
3: $\quad \mathbf{f}_i = \mathbf{p}(I \circ \mathscr{W}(q_{p,t}, c_{p,t}))$ // compute compressed features
4: $\quad (\delta q_{p,t}, \delta c_{p,t}) = R_t(\mathbf{f}_i)$ // evaluate regressor
5: $\quad (q_{p,t}, c_{p,t}) = (q_{p,t-1}, c_{p,t-1}) + (\delta q_{p,t}, \delta c_{p,t})$ // update $(q_{p,t}, c_{p,t})$ parameters
6: $\quad$ Back project parameters to the compressed shape and texture to compute S^t
7: **end for**
8: return fitted shape

4.3.3 Experimental results

Evaluation of the compression rate

We start by testing the compression approach on the MUG database using a cascade of 5 Lasso regressors R^t. First, two bases are evaluated - a Chebychev 3×3 basis using the Chebychev function for calculating the collocation points and the Hermite 10×10 basis using the uniform fonction for collocation points. For each method, six CDAAM model are computed using 7 levels of compression similar to the method used by Stegmann, by keeping only the strongest energy polynomial coefficients. The models are evaluated on all the images from the MUG database, and boxplot results of the error measured as percentage of interocular distance are presented in the figures below.

It can be observed that the results are very stable using both bases. Similar to the conclusions of Stegmann we can observe that the medium error in the standard AAM is

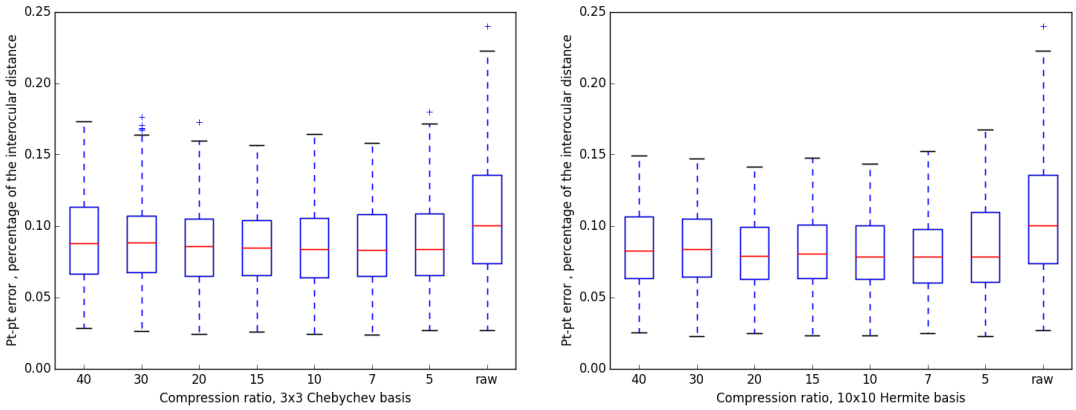


Figure 4.7 Boxplots of alignment error vs compression ratio using Hermite 3×3 and Chebychev 10×10 bases. Wiskers are 1.5 IQR at maximum

worse than all the median values of the compressed 3×3 Hermite basis. Indeed, thanks to the noise suppression during compression the fitting accuracy is improved.

Figures Fig. 4.8 and Fig. 4.9 show the synthesized images computed with the parameters of the compressed AAM.

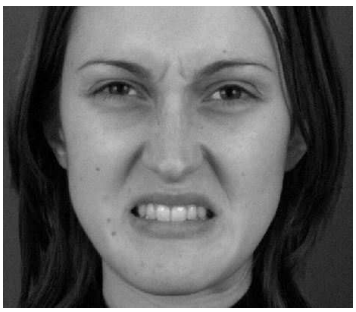


Figure 4.8 Real image

Comparison with the approach using raw pixel information

As we have seen using polynomial bases for compression in AAM improves the quality of the fitting and allows to synthesize a face close to the original.

Tables Tab. 4.2 and Tab. 4.3 show the comparative fitting results of the CDAAM model trained using 5% of data on all the four databases. The CDAAM method using a compression rate of 1:20 outperforms the results of the one using the raw pixel information on Cohn Kanade, Multi Pie and MUG database. When using a compression rate of 1:1, due to the multiscale representation the CDAAM method provides better alignment accuracy, results in accordance with the one presented in the last section.

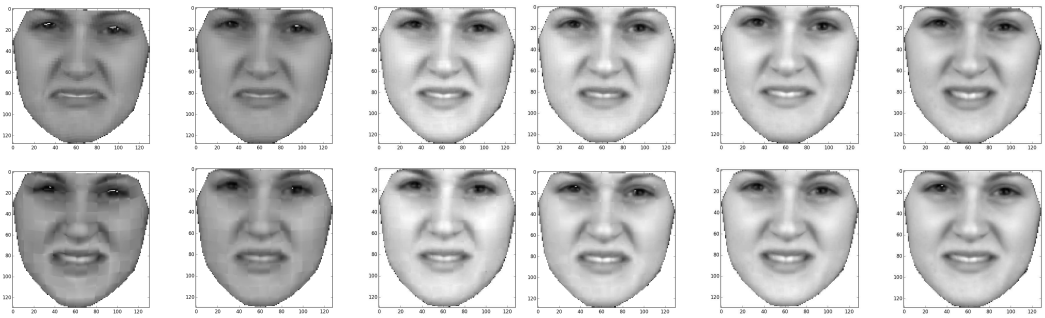


Figure 4.9 Face synthesis using n percents of the coefficients.From left to right compression ratio corresponding to : 40,30,20,15,10,5. On the first line the images are synthesised using Chebychev 3x3 basis and the second one Hermite 10x10 basis.

	CK	**IMM**	**MP**	**MUG**
RAW DAAM	0.169	0.109	0.108	0.093
CDAAM 1:1	**0.058**	**0.092**	**0.083**	**0.070**
CDAAM 1:20	0.147	0.110	0.103	0.077

Table 4.2 Mean error on face interior points

It can also be observed that the error calculated on face interior points, see Fig. 4.2, is consistent with the one measured on the entire set of points.

Figure 4.10 show the comparison of the CDAAM method at 1:1 ratio (using the entire set of polynomial coefficients) with the raw regression AAM. It can be clearly observed that changing the appearance with polynomial coefficients enhance the alignment precision. For example, on MUG database we have 25 percent of points which error is less than 5 percents of interocular distance with the approach using raw intensities, 31 percent with a 1:20 polynomial compression rate and 34 percent of points with the method using 1:1 compression rate.

	CK	**IMM**	**MP**	**MUG**
RAW DAAM	0.183 ± 0.0770	0.132 ± 0.044	0.121 ± 0.050	0.106 ± 0.043
CDAAM 1:1	$\mathbf{0.066} \pm 0.025$	$\mathbf{0.113} \pm 0.033$	$\mathbf{0.098} \pm 0.034$	$\mathbf{0.081} \pm 0.026$
CDAAM 1:20	0.162 ± 0.064	0.135 ± 0.050	0.114 ± 0.043	0.087 ± 0.029

Table 4.3 Mean $\pm$ standard deviation using RAW DAAM and CDAAM methods

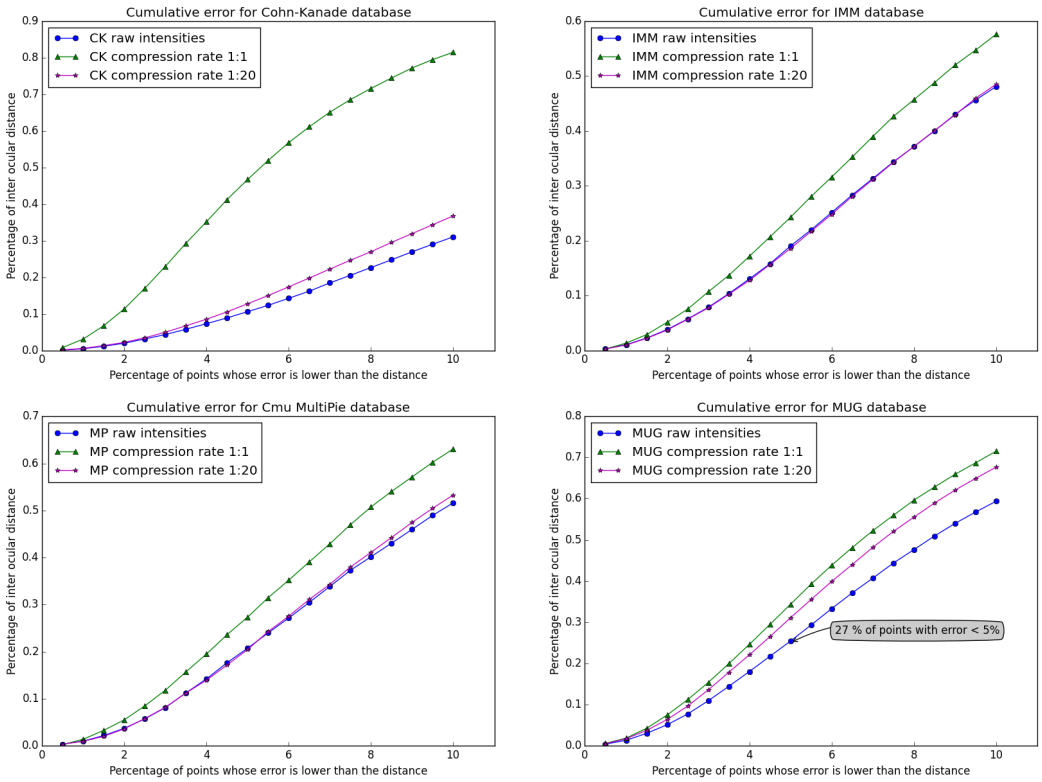


Figure 4.10 Cumulative errors on key points for the various datasets

Comparison with Haar and CDF 9/7 compression method

Similar to the approach in [SFC04] we evaluate the behaviour of Haar wavelet an CDF 9/7 using the CDAAM algorithm. Both of these wavelets were tested using compression ratios in the range 1:5 - 1:40 and compared to the standard raw regression AAM and with the two polynomial basis. The expereiments use three level of wavelet decomposition. Figure 4.11 shows the boxplots of alignment error versus the compression ratio using CDF 9/7 wavelets and Haar wavelets. Unlike [SFC04] conclusions, by using the regression algorithm the average median of CDF wavelets are worse than those of Haar wavelets for all compression rates.

Figure 4.12 shows the average alignment errors with their standard deviation versus the compression ratio using the different approaches. We observe that CDF 9/7 wavelets present the lowest alignment accuracy, yet presenting better results than the raw intensities approach. By using 10×10 polynomial basis and Haar wavelets the alignment results are very close, however for small compression rates the Haar wavelets present better results and for high compression rates the method utilizing polynomial coefficients should be preferred.

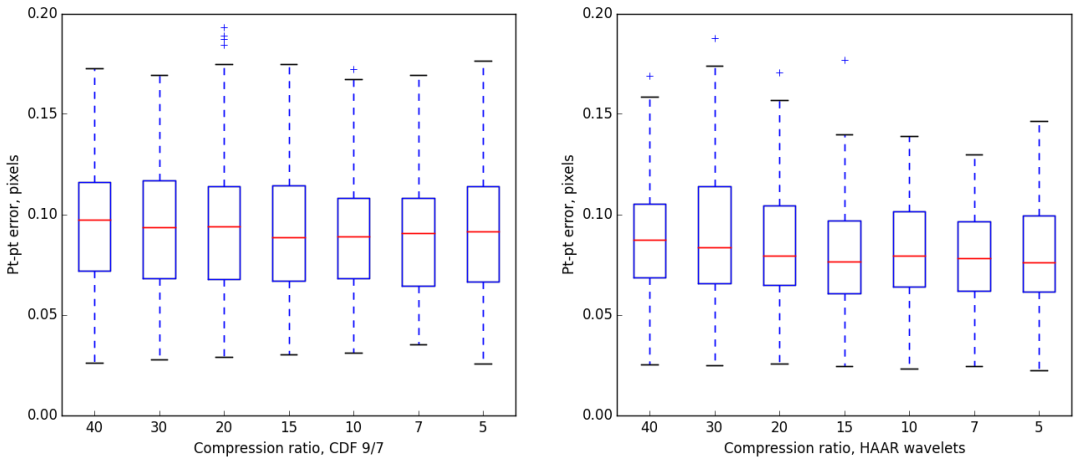


Figure 4.11 Boxplots of alignment error vs compression ratio using CDF 9/7 wavelets and Haar wavelets . Wiskers are 1.5 IQR at maximum

4.4 Discussion and conclusions

In the first part of this chapter, we have proposed two new approaches for texture representation in deformable models, using the AAM framework as an example. Coefficients resulting from polynomial projections of pixel values on a complete basis have been used for image approximation, and compared to global analysis of raw pixel intensities one can usually find in AAM texture models. Experimental results show that our approaches perform very well with face alignment algorithms and depending on the chosen method we obtain robustness to pose changes or facial expression changes. A hybrid approach combining the two proposed methods could be considered to improve the fitting accuracy when the database have multi-pose/multi- expression variations. Because they provide a compact, hierarchical representation of image structures, polynomial decompositions are an efficient alternative to global or redundant texture representations such as Principal Component Analysis on pixel intensities or Gabor wavelets. Moreover, polynomials in the complete basis are orthogonal, which allows a fast computation of texture parameters though direct scalar products within the image.

In the second part we have investigated how an AAM framework can be augmented with polynomial compression to reduce model complexity and have introduced a cascade regression algorithm based on compressed polynomial coefficients. We have carried out experiments using different polynomial bases at seven compression ratios. It was found that using this method alignment accuracy is very stable while increasing compression ratio and keeping a small percentage of data allows to obtain very good alignment results.

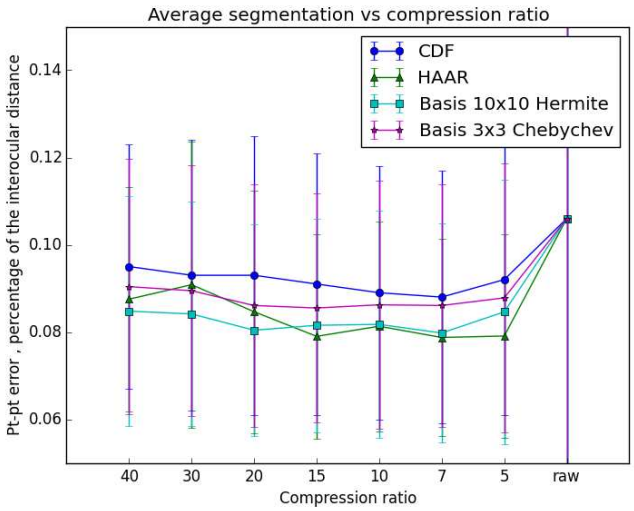


Figure 4.12 Average alignment errors vs compresion ratio. Error bars are on standard error.

We have also compared our approach with Haar wavelet and CDF 9/7 wavelets. Although our method performs as well as the other approaches in terms of alignment accuracy for small compression rates, for high compression rates the method using polynomial coefficients performed consistently better. In addition, we have seen in the Polynomial approximation part 3.3.1 that the polynomial method provided best synthesis quality and is close to Haar wavelets in terms of computational complexity.

In conclusion, we have validated the accuracy and robustness of the polynomial texture apperance on a series of images of the four databases. The results indicate that the polynomial method presents the highly desirable properties of sparsity and compactness that grant good results for alignment.

Chapter 5

Gradient descent approximation using polynomial bases

In the last chapter we have proposed an enhancement for the texture appearance in the AAM framework. However we have seen that the multiresolutional polynomial decomposition approach is equivalent to a filter bank , therefore polynomial coefficients can be used in a gradient descent algorithm.

In this chapter we will take an interest in the generative fitting, review in details the inverse compositional approach and see how we can use polynomials to replace analytically the calculation of gradients.

We showed previously that the ICIA adjustment method for AAMs was introduced in 2004 by Matthews and Baker in [MB04]. They propose to modify the Lucas-Kanade algorithm to make it effective and applicable to AAMs by suggesting an inverse compositional warp update instead of the standard additive update and present an analytical derivation for a gradient descent search.

5.1 Inverse compositional algorithms using polynomials for template matching

Based on forward additive image alignment method, inverse compositional image alignment method is just reversing the roles of the trained model template image $\mathbf{A}(\mathbf{x})$ and the input image $I(\mathbf{W}(\mathbf{W}(\mathbf{x};\delta r);r))$, which results in :

$$\sum_{\mathbf{x}} [A(\mathbf{W}(\mathbf{x},\delta\mathbf{r})) - I(\mathbf{W}(\mathbf{x},\mathbf{r}))]^2 \tag{5.1}$$

After taking the 1-order Taylor series expansion in terms of $\delta\mathbf{r}$ at $\delta\mathbf{r} = \mathbf{0}$ we have:

$$\begin{aligned}&\sum_{\mathbf{x}}\left[A\left(\mathbf{W}\left(\mathbf{x},\mathbf{0}\right)\right)+\frac{\partial A(\mathbf{W}(\mathbf{x},\delta\mathbf{r}))}{\partial(\mathbf{W}(\mathbf{x},\delta\mathbf{r}))}\frac{\partial(\mathbf{W}(\mathbf{x},\delta\mathbf{r}))}{\partial\delta\mathbf{r}}\,|_{\delta\mathbf{r}\to 0}(\delta\mathbf{r}-0)-I\left(\mathbf{W}\left(\mathbf{x},\mathbf{r}\right)\right)\right]^2\\&=\sum_{\mathbf{x}}\left[A\left(\mathbf{x}\right)-\nabla A\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\delta\mathbf{r}-I\left(\mathbf{W}\left(\mathbf{x},\mathbf{r}\right)\right)\right]^2\end{aligned} \tag{5.2}$$

where ∇A is the gradient image for the template image.

We then minimize 5.2 by computing its partial derivative in terms of δr and find the next solution :

$$\delta\mathbf{r} = \mathbf{H}\left[I\left(\mathbf{W}\left(\mathbf{x},\mathbf{r}\right)\right)-A\left(\mathbf{x}\right)\right] \tag{5.3}$$

with

$$\mathbf{H} = \left(\left(\nabla A\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\right)^T\left(\nabla A\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\right)\right)^{-1}\left(\nabla A\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\right)^T \tag{5.4}$$

We remind the inverse compositional algorithm as presented by Matthews and Baker in Alg 5 :

Algorithm 5 The Inverse Compositional Algorithm

Pre-compute:

(3) Evaluate the gradient ∇A of the template $A\left(\mathbf{x}\right)$
(4) Evaluate the Jacobian $\frac{\partial\mathbf{W}}{\partial\mathbf{r}}$ at $\left(\mathbf{x};\mathbf{0}\right)$
(5) Compute the steepest descent images $\nabla A\frac{\partial\mathbf{W}}{\partial\mathbf{r}}$
(6) Compute the Hessian matrix

Iterate:

(1) Warp I with $\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)$ to compute $I\left(\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)\right)$
(2) Compute the error image $I\left(\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)\right)-A\left(\mathbf{x}\right)$
(7) Compute $\sum_{\mathbf{x}}\left[\nabla A\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\right]^T\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)\right)-A\left(\mathbf{x}\right)\right]$
(8) Compute $\Delta\mathbf{r}$
(9) Update the warp $\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)\leftarrow\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)\circ\mathbf{W}\left(\,\mathbf{x};\Delta\mathbf{r}\right)^{-1}$

until $\|\Delta\mathbf{r}\|\leq\varepsilon$

The function $\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)$ is the parametrized set of allowed warps where $\mathbf{r}=\left(r_1\ldots r_n\right)^{\mathrm{T}}$ is a vector of parameters. It takes pixel $\mathbf{x}$ in the coordinate frame of a template and maps it to the sub pixel location $\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)$ in the coordinate frame of the image I. As we want to track an image patch moving in 3D we will consider the set of affine warps :

$$\mathbf{W}\left(\mathbf{x};\mathbf{r}\right) = \begin{pmatrix}1+r_1 & r_2 & r_3\\ r_4 & 1+r_5 & r_6\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix} \tag{5.5}$$

In equation 5.5, r_3 and r_6 corresponds to translation, r_1 and r_5 to scaling and r_2 and r_4 to shear. If we compute the Jacobian and Hessian of the warp we obtain :

$$\frac{\partial \mathbf{W}}{\partial \mathbf{r}} = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial^2 \mathbf{W}}{\partial \mathbf{r}^2} = \mathbf{0} \tag{5.6}$$

We will first show now how to use polynomials in the algorithm and what are its advantages. The use of the polynomial basis is similar to the convolution with a filter bank that allows to extract the different frequency components of a signal and offers the possibility to use multiresolution piecewise polynomial decomposition. Our representation is non redundant - if the signal is represented with N samples, the "wavelet-like" polynomial representation contains at maximum N coefficients. Moreover, this representation is invertible: a perfect reconstruction of the original signal from the coefficients.

Since we can consider the projection on a polynomial basis $\mathbf{B}_{i,j}$ as a multi-scale finite differences operator $\partial_1^i \partial_2^j$ we propose to replace the gradients calculated in the algorithm by the ones obtained by polynomial projections. We will use a Legendre 3×3 polynomial basis whose coefficients are given in Table5.1

B00	0.333	0.333	0.333	**B01**	-0.408	-0.408	-0.408	**B02**	0.235	0.235	0.235
	0.333	0.333	0.333		0	0	0		-0.471	-0.471	-0.471
	0.333	0.333	0.333		0.408	0.408	0.408		0.235	0.235	0.235
B10	-0.408	0	0.408	**B11**	0.5	0	-0.5	**B12**	-0.288	0	0.288
	-0.408	0	0.408		0	0	0		0.577	0	-0.577
	-0.408	0	0.408		-0.5	0	0.5		-0.288	0	0.288
B20	0.235	-0.471	0.235	**B21**	-0.288	0.577	-0.288	**B22**	0.166	-0.333	0.166
	0.235	-0.471	0.235		0	0	0		-0.333	0.666	-0.333
	0.235	-0.471	0.235		-0.288	0.577	-0.288		0.166	-0.333	0.166

Table 5.1 Polynomial coefficients for a 3×3 Legendre basis

The projections obtained by the bases are undersampled compared to the initial image, we therefore start our algorithm by projecting the template and the input image in the polynomial basis. Projecting these two images in the B_{00} polynomial is similar to a filtering operation with a mean filter followed by a subsampling. It can be noticed that the polynomials B_{01} and B_{10} present strong similarities with Gx and Gy gradient filters. With respect to standard representations, the basis filters are reversed spatially on the horizontal axis, so the results acquired following a projection on the B_{01} are multiplied by -1.

We include the polynomial coefficients in the inverse compositional algorithm by replacing the computation of part 3 of the Alg 5 with projections of polynomial basis, and those

of parts 5,6,1,2 and 7 respectively (due to undersampling) and the obtained results are very close to those received without polynomials (see Table 5.2) with execution times divided roughly by 10. Therefore we deduce that projections on polynomials bases B_{01} and B_{10} are a good alternative to conventional filters for calculating the gradients of an image.

We present the results of a face template alignment on 4 images of two subjects from the MUG database (see Figure 5.1). Each input image is deformed twice with random values

Input image Template Input image Template

Figure 5.1 Images used for template alignement

and then we use the inverse compositional algorithm to align the face region to the template image on witch we applied some blurring and noise. The input regions for each case, the noisy templates and the results from alignment are presented in the figure 5.2.

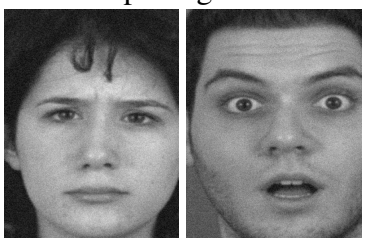
	Image1		Image2		Image3		Image4	
	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
ICIA	17.719	16.232	17.703	15.994	17.301	15.559	17.158	18.722
ICIA Polynomes	17.695	**2.397**	17.695	**2.100**	17.279	**2.047**	17.048	**2.162**

Table 5.2 Comparative results for face template matching.

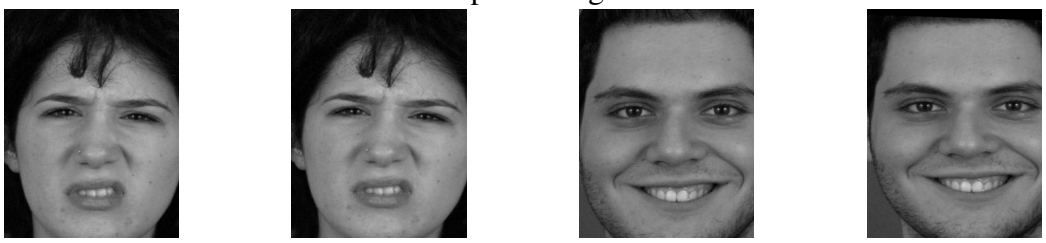
It can be observed in tab 5.2 that PSNR values are rather low. This is explained by the added noise and blur to the input image. Using the polynomial coefficients is very efficient, as our method is 7x times faster, obtaining very close PSNR values.

From Table 5.1 it can be observed that using the last algorithm; after projection on polynomial basis there are still a part of unused data (since we use only the B_{00}, B_{10} ans B_{01} polynomial coefficients). Hence we implement the inverse compositional algorithm using Newton's algorithm, which uses a second order Taylor expansion and therefore uses second order derivatives of the image and template (already calculated with projections on a polynomial basis) in the gradient descent approximation.

Deformed input images, in white the input regions used for alignment with the template



Template images



Resulting images, aligned on the templates

Figure 5.2 Images used for template alignement

In the Newton algorithm, the Hessian of the cost function differs from the Hessian of the Gauss Newton algorithm (see 5.4) by: $\frac{\partial^2 G}{\partial \mathbf{r}^2} =$

$$\left(\left[\frac{\partial \mathbf{W}}{\partial \mathbf{r}} \right]^T \left[\frac{\partial^2 A}{\partial \mathbf{x}^2} \right] \left[\frac{\partial \mathbf{W}}{\partial \mathbf{r}} \right] + \nabla A \left[\frac{\partial^2 \mathbf{W}}{\partial \mathbf{r}^2} \right] \right) [A(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{r}))] + \left[\nabla A \frac{\partial \mathbf{W}}{\partial \mathbf{r}} \right]^T \left[\nabla A \frac{\partial \mathbf{W}}{\partial \mathbf{r}} \right] \tag{5.7}$$

This approximation is more complex as it requires the second order derivatives of the warp and the template and can not be precomputed as it depends on the parameters $\mathbf{r}$ through $I(\mathbf{W}(\mathbf{x};\mathbf{r}))$. However it is straightforward to replace the second order derivatives of the template with the results obtained from the polynomial projections. As in the Gauss Newton approach using polynomials, the calculations will be performed on undersampled images hence they'll be faster.

The results obtained using Newton method are presented in Figure 5.3. We can observe that the PSNR results are slightly better, with greater execution times.

	Image1		Image2		Image3		Image4	
	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
ICIA	17.731	15.460	17.742	22.155	17.318	23.648	17.165	31.308
ICIA Polynomes	17.696	**3.565**	17.697	**3.593**	17.297	**3.556**	17.048	**3.692**

Table 5.3 Comparative results for face template matching.

5.2 Polynomial inverse compositional algorithm for AAMs

We presented in the last section how the inverse compositional image alignment algorithm minimizes the error between an input image $I(\mathbf{x})$ and a constant template image $A(\mathbf{x})$.

We reformulate the inverse compositional algorithm to entertain fitting across multiple polynomial filter responses. The error functions can be written as,

$$\sum_{\mathbf{x}} \left[\left\{ \left\langle I\left(\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)\right) \middle| B_{i,j} \right\rangle_w \right\}_{i=1}^{D} - \left\{ \left\langle A\left(\mathbf{W}\left(\mathbf{x};\Delta\mathbf{r}\right)\right) \middle| B_{i,j} \right\rangle_w \right\}_{i=1}^{D} \right]^2 \tag{5.8}$$

Where $B_{i,j}$ is the i,j polynomial of the basis of size D, and $\{.\}_{i=1}^{D}$ represents the concatenation operation i.e. $\{x_i\}_{i=1}^{D} = \left[x_1^T ... x_D^T\right]^T$

The error in equation 5.8 can equivalently be written as:

$$\sum_{\mathbf{x}} \left[\left\{ \left\langle \left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{r}\right)\right) - A\left(\mathbf{W}\left(\mathbf{x};\Delta\mathbf{r}\right)\right) \right] \middle| B_{i,j} \right\rangle_w \right\}_{i=1}^{D} \right]^2 \tag{5.9}$$

From this equation we see that the polynomial representation can be directly used in the Lucas Kanade framework. A key element of the error function in equation 5.8 is that solving this function using Lucas Kanade strategy requires the linearisation of:

$$\left\langle A\left(\mathbf{W}\left(\mathbf{x};\Delta\mathbf{r}\right)\right) \middle| B_{i,j} \right\rangle \approx \left\langle A\left(\mathbf{W}\left(\mathbf{x};0\right)\right) \middle| B_{i,j} \right\rangle + \frac{\left\langle \partial A\left(\mathbf{W}\left(\mathbf{x};\Delta\mathbf{r}\right)\right) \middle| B_{i,j} \right\rangle}{\partial \mathbf{r}} \Delta\mathbf{r} \tag{5.10}$$

where

$$\begin{aligned} &\frac{\left\langle \partial A(\mathbf{W}(\mathbf{x};\Delta\mathbf{r})) \middle| B_{i,j} \right\rangle}{\partial \mathbf{r}} \Delta\mathbf{r} = \frac{\left\langle \partial A(\mathbf{W}(\mathbf{x};\Delta\mathbf{r})) \middle| B_{i,j} \right\rangle}{\partial \mathbf{W}(\mathbf{x};\Delta\mathbf{r})} \frac{\partial \mathbf{W}(\mathbf{x};\Delta\mathbf{r})}{\partial \Delta\mathbf{r}} \Delta\mathbf{r} = \\ &= \frac{\left\langle \partial A(\mathbf{W}(\mathbf{x};0)) \middle| B_{i,j} \right\rangle}{\partial \mathbf{W}(\mathbf{x};0)} \frac{\partial \mathbf{W}(\mathbf{x};\mathbf{r})}{\partial \mathbf{r}} \Delta\mathbf{r} = \left\langle \nabla A \middle| B_{i,j} \right\rangle \frac{\partial \mathbf{W}(\mathbf{x};\mathbf{r})}{\partial \mathbf{r}} \Delta\mathbf{r} \end{aligned}$$

such that $\frac{\partial \mathbf{W}(\mathbf{x};0)}{\partial \mathbf{r}}$ is the Jacobian of the warp function. We evaluate the Jacobian of the template image with respect to the x and y coordinates directly by re-projecting the template polynomial representation on the polynomials of degree 1 calculated in a Hermite basis $\mathbb{B} = \left\{ B_{i,j}(x) \right\}_{\substack{i=0..d_1 \\ j=0..d_2}}$:

$$\left\langle \nabla A \left| B_{i,j} \right. \right\rangle = \left\langle \left(\left\langle A \left| B_{1,0} \right. \right\rangle, \left\langle A \left| B_{0,1} \right. \right\rangle \right) \left| B_{i,j} \right. \right\rangle \tag{5.11}$$

5.2.1 First order Taylor approximation using complete polynomial basis

The Active Polynomial Models proposed in this work are an extention of the AAM revisited and are designed to use the same shape and motion model as the ones used by AAMs but have a different appearance model and cost function (see Eq. 5.12) to fit this model.

Given a generic AAM and a video frame I at time t, we propose an approach that uses the following cost function to perform the face model fitting:

$$\sum_{\mathbf{x}} \sum_{i,j} \left[\left\langle I \left| B_{i,j} \right. \right\rangle_{\omega} (\mathbf{W}(\mathbf{x};\mathbf{r})) - \left\langle A \left| B_{i,j} \right. \right\rangle_{\omega} (\mathbf{W}(\mathbf{x};\Delta\mathbf{r})) \right]^2 \tag{5.12}$$

The first order Taylor approximation for the inverse compositional algorithm is known as the Gauss Newton approximation, and since we use polynomials in the calculations we will call our method Gauss Newton Polynomial ICIA.

Using the polynomial bases instead of minimizing the sum of squares differences between a constant template $A_0(\mathbf{x})$ and an example image $I(\mathbf{x})$ with respect to the warp parameters $\mathbf{r}$ and $\mathbf{q}$ (global transform parameters) we will minimize the difference between an example image and the corresponding template calculated by projections into the complete polynomial basis.

Therefore we will use the next algorithm:

Algorithm 6 Polynomial Inverse Compositional Algorithm with Appearance Variation and Global Shape Transform

Pre-compute:

(i) Generate and project the appearance into the complete basis. The appearance part of the model will be calculated using Equation 4.6

(3) Evaluate the gradient $\nabla\overline{\mathbf{A_p}}$ of the template $\overline{\mathbf{A_p}}(\mathbf{x})$ by projections on the 1st an 2nd polynomials of the basis using equation 5.13

(4) Evaluate the Jacobians $\frac{\partial \mathbf{N}}{\partial \mathbf{q}}$ and $\frac{\partial \mathbf{W}}{\partial \mathbf{r}}$ at $(\mathbf{x};\mathbf{0})$ where $\frac{\partial \mathbf{N}}{\partial \mathbf{q}}$ is the global shape transform

(5) Compute the modified steepest descent images using Equations 5.15 and 5.14

(6) Compute the Hessian matrix

Iterate:

(ii) Project the image I in the complete basis $\rightarrow I_p$

(1) Warp I_p with $\mathbf{W}(\mathbf{x};\mathbf{r})$ followed by $\mathbf{N}(\mathbf{x};\mathbf{q})$ to compute $I_p(\mathbf{N}(\mathbf{W}(\mathbf{x};\mathbf{r});\mathbf{q}))$

(2) Compute the error image $I_p(\mathbf{N}(\mathbf{W}(\mathbf{x};\mathbf{r});\mathbf{q})) - \overline{\mathbf{A_p}}(\mathbf{x})$

(7) Compute $\sum_{\mathbf{x}\in\mathbf{s_0}} SD_i(\mathbf{x})\left[I(\mathbf{N}(\mathbf{W}(\mathbf{x};\mathbf{r});\mathbf{q})) - \overline{\mathbf{A_p}}(\mathbf{x})\right]$ for $i = 1,\ldots,n+4$

(8) Compute$(\Delta\mathbf{q},\Delta\mathbf{r})$ by multiplying the resulting vector by the inverse Hessian

(9) Update $(\mathbf{N}\circ\mathbf{W})(\mathbf{x};\mathbf{q},\mathbf{r}) \leftarrow (\mathbf{N}\circ\mathbf{W})(\mathbf{x};\mathbf{q},\mathbf{r}) \circ (\mathbf{N}\circ\mathbf{W})(\mathbf{x};\Delta\mathbf{q},\Delta\mathbf{r})^{-1}$

The gradient $\nabla\overline{\mathbf{A_p}}$ of the template $\overline{\mathbf{A_p}}$ is evaluated by projections on the first order polynomials of the basis $B_{1,0}$ and $B_{0,1}$:

$$\nabla\overline{\mathbf{A}}_\mathbf{p} = \left(\sum\left\langle\overline{\mathbf{A}}_\mathbf{p}\middle|B_{1,0}\right\rangle_w, \sum\left\langle\overline{\mathbf{A}}_\mathbf{p}\middle|B_{0,1}\right\rangle_w\right) \text{ where } \overline{\mathbf{A}}_\mathbf{p} = \begin{bmatrix} \overline{B_{0,0}} \\ \overline{B_{0,1}} \\ \vdots \\ \overline{B_{d_1,d_2}} \end{bmatrix} \tag{5.13}$$

With respect to the equations of Matthews and Baker the modified steepest descent images that are used in our approach are computed using:

$$SD_{j,r}(\mathbf{x}) = \nabla\bar{\mathbf{A}}_\mathbf{p}\frac{\partial \mathbf{N}}{\partial \mathbf{q}_j} - \sum_{i=1}^{r}\left(\sum_{\mathbf{x}\in\mathbf{s}}\lambda_p(\mathbf{x})\nabla\bar{\mathbf{A}}_\mathbf{p}\frac{\partial \mathbf{N}}{\partial \mathbf{q}_j}\right) \tag{5.14}$$

for each of the similarity parameters and :

$$SD_{j+4,r}(\mathbf{x}) = \nabla\bar{\mathbf{A}}_\mathbf{p}\frac{\partial \mathbf{W}}{\partial \mathbf{r}_j} - \sum_{i=1}^{r}\left(\sum_{\mathbf{x}\in\mathbf{s}}\lambda_p(\mathbf{x})\nabla\bar{\mathbf{A}}_\mathbf{p}\frac{\partial \mathbf{W}}{\partial \mathbf{r}_j}\right) \tag{5.15}$$

for $\mathbf{r}$ where $j = 1..n$. It can be observed that we use in these equation the appearance polynomial parameters λ_p joined with polynomial gradients.

Alignment results

We present the comparison result of face alignment using the algorithm presented by Matthews and Baker and its polynomial version in Table 5.4.

	CK	**IMM**	**MP**	**MUG**
Gauss-Newton ICIA	0.175 ± 0.139	0.418 ± 0.160	$\mathbf{0.160} \pm 0.193$	0.148 ± 0.094
Gauss Newton Polynomial ICIA	$\mathbf{0.165} \pm 0.110$	$\mathbf{0.194} \pm 0.058$	0.245 ± 0.081	$\mathbf{0.123} \pm 0.054$

Table 5.4 Mean $\pm$ standard deviation using ICIA and Polynomial ICIA methods

The results show that the method using polynomials for the gradient descent and for texture representation performs better than the original algorithm for Cohn Kanade, IMM for which the error is greatly improved and for MUG database. For Cohn Kanade, MultiPie and MUG databases, where the faces are not having rotations, the original algorithm shows good results, and for IMM, where each face has different orientations the algorithms performs poorly. By using polynomials, we can obtain good results even for images presenting rotations.

In Table 5.5 are presented the alignment results calculating the error on face interior points. They are consistent with the one presented before, as we obtain smaller errors using polynomials for all databases except Multi Pie.

	CK	**IMM**	**MP**	**MUG**
Gauss-Newton ICIA	**0.152**	0.328	**0.139**	0.139
Gauss Newton Polynomial ICIA	0.155	**0.153**	0.224	**0.114**

Table 5.5 Mean error on face interior points

5.2.2 Second order Taylor approximation using bases

Matthews and Baker do not recommend to use the full Newton Hessian method because this approach uses a sophisticated estimate of the Hessian that is presumed noiseless. They also state that the increased noise in estimating the second order derivatives of the template outweights the increased sophistication in the algorithm.

Projections in the polynomial basis include convolution with the weighting functions used for basis construction. Using the Hermite basis we will convolve the input image with a gaussian filter therefore we will limit the noise in the gradient and second derivatives. Therefore we propose the Newton approach using projections on the first and second order basis polynomials. The Newton inverse compositional AAM fitting algorithm including global shape transform is summarized in Alg 7.

Algorithm 7 Newton Inverse Compositional Algorithm with Global Shape Transform (modified version of [BM02])

Pre-compute:

(i) Generate and project the appearance into the complete basis. The appearance part of the model will be calculated using Equation 4.6

(3) Evaluate the gradient $\nabla\overline{\mathbf{A_p}}$ and the second derivatives $\frac{\partial^2\overline{\mathbf{A_p}}}{\partial\mathbf{x}^2}$ of the template $\bar{\mathbf{r}}(\mathbf{A_p})$

(4) Evaluate the Jacobians $\frac{\partial\mathbf{N}}{\partial\mathbf{q}}$, $\frac{\partial\mathbf{W}}{\partial\mathbf{r}}$ and the Hessians $\frac{\partial^2\mathbf{W}}{\partial\mathbf{r}^2}$, $\frac{\partial^2\mathbf{N}}{\partial\mathbf{q}^2}$ at $(\mathbf{x};\mathbf{0})$

(5) Compute $\nabla\overline{\mathbf{A_p}}\frac{\partial\mathbf{W}}{\partial\mathbf{r}}$, $\left[\nabla\overline{\mathbf{A_p}}\frac{\partial\mathbf{W}}{\partial\mathbf{A_p}}\right]^T\left[\nabla\overline{\mathbf{A_p}}\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\right]$, $\left[\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\right]^T\left[\frac{\partial^2\overline{\mathbf{A_p}}}{\partial\mathbf{x}^2}\right]\left[\frac{\partial\mathbf{W}}{\partial\mathbf{r}}\right]+\nabla\overline{\mathbf{A_p}}\frac{\partial^2\mathbf{W}}{\partial\mathbf{r}^2}$

and $\nabla\overline{\mathbf{A_p}}\frac{\partial\mathbf{N}}{\partial\mathbf{q}}$, $\left[\nabla\overline{\mathbf{A_p}}\frac{\partial\mathbf{N}}{\partial\mathbf{q}}\right]^T\left[\nabla\overline{\mathbf{A_p}}\frac{\partial\mathbf{N}}{\partial\mathbf{q}}\right]$, $\left[\frac{\partial\mathbf{N}}{\partial\mathbf{q}}\right]^T\left[\frac{\partial^2\overline{\mathbf{A_p}}}{\partial\mathbf{x}^2}\right]\left[\frac{\partial\mathbf{N}}{\partial\mathbf{q}}\right]+\nabla\overline{\mathbf{A_p}}\frac{\partial^2\mathbf{N}}{\partial\mathbf{q}^2}$

Iterate:

(ii) Project the image I in the complete basis $\rightarrow I_p$

(1) Warp I_p with $\mathbf{W}(\mathbf{x};\mathbf{r})$ followed by $\mathbf{N}(\mathbf{x};\mathbf{q})$ to compute $I_p(\mathbf{N}(\mathbf{W}(\mathbf{x};\mathbf{r});\mathbf{q}))$

(2) Compute the error image $I_p(\mathbf{N}(\mathbf{W}(\mathbf{x};\mathbf{r});\mathbf{q}))-\overline{\mathbf{A_p}}(\mathbf{x})$

(6) Compute the Hessian matrix $\sum_{\mathbf{x}}\frac{\partial^2\mathbf{G}}{\partial\mathbf{r}^2}$ using Equation 5.16

(7) Compute $\left[\sum_{\mathbf{x}}\frac{\partial\mathbf{G}}{\partial\mathbf{r}}\right]_p^T=\sum_{\mathbf{x}\in\mathbf{s_0}}SD_i(\mathbf{x})\left[I_p(\mathbf{N}(\mathbf{W}(\mathbf{x};\mathbf{r});\mathbf{q}))-\overline{\mathbf{A_p}}(\mathbf{x})\right]$ for $i=1,...,n+4$

(8) Compute$(\Delta\mathbf{q},\Delta\mathbf{r})$ by multiplying the resulting vector by the inverse Hessian

(9) Update $(\mathbf{N}\circ\mathbf{W})(\mathbf{x};\mathbf{q},\mathbf{r})\leftarrow(\mathbf{N}\circ\mathbf{W})(\mathbf{x};\mathbf{q},\mathbf{r})\circ(\mathbf{N}\circ\mathbf{W})(\mathbf{x};\Delta\mathbf{q},\Delta\mathbf{r})^{-1}$

The Hessian matrix can be calculated using :

$$\frac{\partial^2\mathbf{G}}{\partial\mathbf{r}^2}=\left(\left[\frac{\partial\mathbf{J}}{\partial\mathbf{r}}\right]^T\left[\frac{\partial^2\bar{\mathbf{A}}_\mathbf{p}}{\partial\mathbf{x}^2}\right]\left[\frac{\partial\mathbf{J}}{\partial\mathbf{r}}\right]+\nabla\bar{\mathbf{A}}_\mathbf{p}\frac{\partial^2\mathbf{J}}{\partial\mathbf{r}^2}\right)\left(I_p(\mathbf{N}(\mathbf{W}(\mathbf{x};\mathbf{r});\mathbf{q}))-\bar{\mathbf{A}}_\mathbf{p}(\mathbf{x})\right)+\left[\nabla\bar{\mathbf{A}}_\mathbf{p}\frac{\partial\mathbf{J}}{\partial\mathbf{r}}\right]^T\left[\nabla\bar{\mathbf{A}}_\mathbf{p}\frac{\partial\mathbf{J}}{\partial\mathbf{r}}\right] \tag{5.16}$$

where $\frac{\partial\mathbf{J}}{\partial\mathbf{r}}=\begin{bmatrix}\frac{\partial\mathbf{N}}{\partial\mathbf{q}}\\ \frac{\partial\mathbf{W}}{\partial\mathbf{r}}\end{bmatrix}$ and $\overline{\mathbf{A_p}}=\begin{bmatrix}\overline{B_{0,0}}\\ \overline{B_{0,1}}\\ \vdots\\ \overline{B_{d_1,d_2}}\end{bmatrix}$

In our experiments the second order derivatives are calculated via projections on polynomial basis and $\frac{\partial^2 \mathbf{J}}{\partial \mathbf{r}^2} = 0$

Alignment experiments

We compared our method with the Newton approach on each database. After training two models per database - one using the Newton approach and one using the Newton polynomial approach on 30 images , we fit the resulting models on the resting images. The average error and the standard deviation are displayed in Table 5.6.

	CK	**IMM**	**MP**	**MUG**
Newton	0.080 ± 0.030	0.159 ± 0.066	0.108 ± 0.053	0.086 ± 0.033
Newton Polynomial ICIA	$\mathbf{0.060} \pm 0.018$	$\mathbf{0.109} \pm 0.043$	$\mathbf{0.081} \pm 0.026$	$\mathbf{0.065} \pm 0.017$

Table 5.6 Mean $\pm$ standard deviation using Newton and Polynomial Newton methods

First, it can be observed that the average errors using the Newton method are significantly lower than the ones using the Gauss Newton method (see Table 5.4). Second, as previously, by using polynomial projections we obtain better alignment results for all databases. The next table Tab.5.7 shows that results on interior points are consistent with the one measured on the entire set of points, as we obtain smaller errors using polynomials. We can also conclude that among the four databases face interior points are most precisely calculated on Cohn Kanade and MUG database.

	CK	**IMM**	**MP**	**MUG**
Newton ICIA	0.070	0.120	0.091	0.072
Newton Polynomial ICIA	**0.049**	**0.077**	**0.065**	**0.051**

Table 5.7 Mean error on face interior points

Figure 5.3 presents the cumulative percentage of points which error is lower than a certain percentage of the inter ocular distance. It can be observed that the best results are obtained for Cohn Kanade database, followed by the MUG and Multi Pie datasets using our polynomial method. The gain provided by the use of polynomials is substantial for all datasets and overcomes the small extra time spent for polynomial projections.

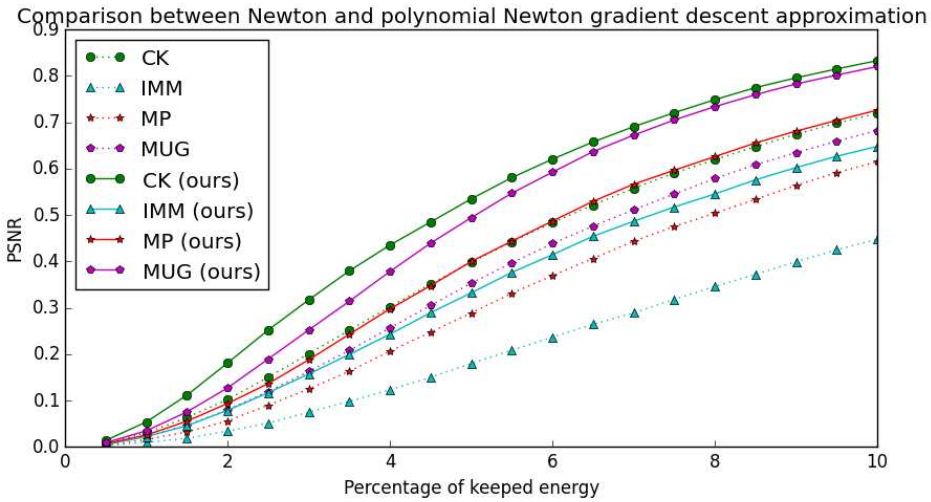


Figure 5.3 Cumulative errors on key points for the various datasets using the Newton (dotted line) and polynomial Newton (solid line) algorithms

5.3 Discussion and conclusion

We have presented in this chapter two extensions of the inverse compositional image alignment algorithm using polynomial projections. To our knowledge we propose the first unified solution dealing with gradient descent and texture representation into a single consistent model.

The algorithms have been evaluated on challenging datasets, including Multi-PIE alignment accuracy that focuses on accessing the performance with combined identity, pose, expression and illumination variation.

We believe that the framework is a solid basis to explore more complex facial models, which we suspect may even further improve alignment quality in the images/videos in the facial alignment context.

Overview on polynomial AAMs

In this chapter we have proposed to use polynomial projection coefficients in different parts of the Active Appearance algorithm.

First, we replaced the texture representation model by approximating image structures with polynomial projections into an orthonormal basis. Two different models were proposed and compared to the raw image intensity representations. Experimental results show that the two approaches perform well with face alignment algorithms and according to the chosen method, robustness to pose changes or facial expression changes is acquired.

Next, we have proposed an enhanced appearance model where subsets of polynomial coefficients were used in order to obtain different compression ratios. In addition to a detailed review of the method and a discussion of their integration in a regression framework we compared our polynomial basis with HAAR and CDF 9/7 wavelets. It was shown also that using a polynomial representation at compression ratio 1:1, due to the multiscale representation of the data, alignment results are more accurate than a conventional AAM, confirming the results of the first part of the chapter. We have also shown that at higher compression ratios our method presents a decreased complexity and a better alignment accuracy.

Then we showed that the polynomial coefficients can be used in the gradient descent algorithm and reformulated the inverse compositional algorithm to entertain fitting across multiple polynomial filter responses. We demonstrated that coefficients obtained from polynomial projections on the polynomials of degree 1 can be used in the Gauss Newton descent algorithm and the ones obtained on the first and second order polynomials in the Newton approach.

The figure 5.4 presents a summary comparison of all presented methods in this chapter on MUG database. It can be observed that for frontal images, presenting facial expressions the best method is the one using Newton polynomial gradient algorithm as we have more than 80% of the points detected with this method which error is less than 10 % of the inter ocular distance calculated on the database. This method is followed by the one using regression and the worse results are obtained with the Gauss-Newton gradient descent algorithm.

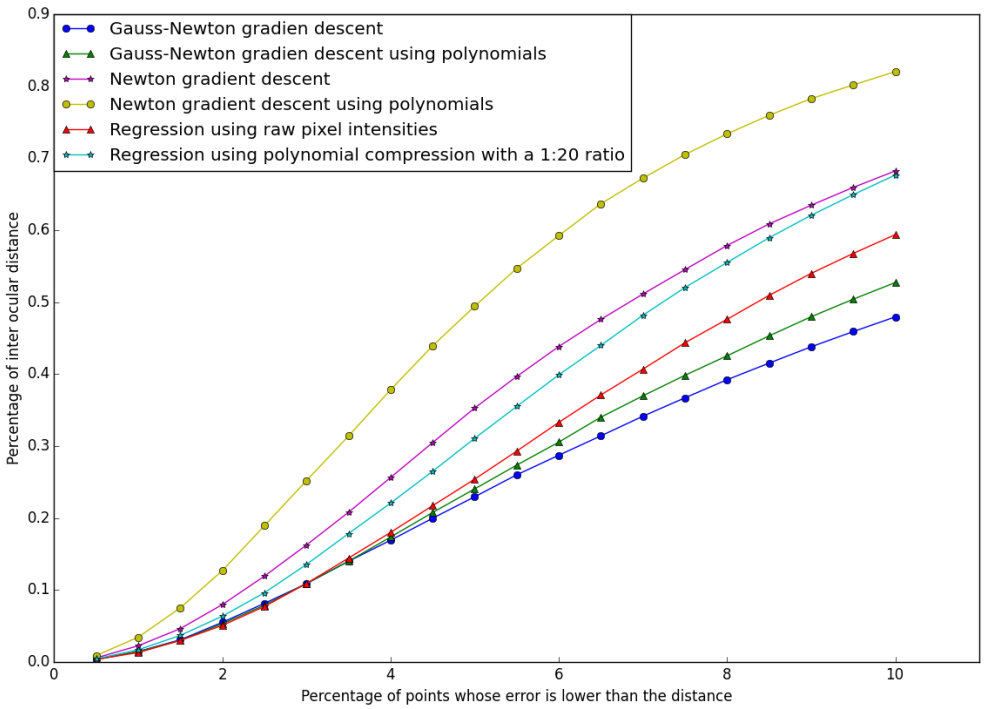


Figure 5.4 Comparison of the cumulative error on keypoints points on MUG database using different approaches

However, due to computational complexity of the Newton method (close to 45 seconds per image) the bests results can be considered as the one using compressed polynomial regression as we obtain more than 60 % of the point with an acceptable error and adequate execution time (0.07 seconds per image).

We have seen that the use of polynomial projection coefficients can be used in different parts of the Active Appearance algorithm : the texture representation or the gradient descent algorithm . We will see in the next part how polynomial bases can be used for interesting points and areas detection and as a descriptor for facial expression recognition.

From points of interest to facial expressions

Chapter 6

Points of interest detection in the characterization of facial textures

6.1 Introduction

The purpose of a keypoint detector is to determine the points in an image that are relevant enough to allow an efficient object description and correspondence with respect to point-of-view variations and to provide a limited set of well localized and individually identifiable points.

Detecting regions covariant with a class of transformations in an image has now reached some maturity in the computer vision literature. Several categories of keypoint detectors have been proposed over the years: corner detectors (found at various types of junctions, on highly textured surfaces, at occlusion boundaries, such as Harris [HS88], and Susan [SB97] detector) and blob detectors (characterized by their boundaries such as SIFT [Low99], SURF [BTVG06], MSER [DB06]). They are widely used in many computer vision applications such as 3D reconstruction, motion tracking, robot navigation, object recognition, image alignment (panoramas) and various methods have been proposed during the past decade.

Considering that polynomial bases have been used to detect and characterize singularities in a vector field we propose to use them for an accurate keypoint localization in the image domain.

6.2 Singular points detection of color/grayscale images

We consider a singular point of a color image any singular point (or singularity) of the normal field to the image surface.

6.2.1 Vector field of normals

If an image is noted as a vector application from $\mathbb{R}^2$ to $\mathbb{R}^3$:

$$\mathbf{I} : \mathbf{x} = (x_i)_{i\in\{1,2\}} \mapsto \mathbf{I}(\mathbf{x}) = (I_j(\mathbf{x}) = I_j(x_1, x_2))_{j\in\{1,2,3\}} \tag{6.1}$$

then, for each point, the normal vector to the geodesic of the surface, more commonly known as the normal vector is given by :

$$\eta = \left(\sum_j \partial_1 I_j, \sum_j \partial_2 I_j \right) \tag{6.2}$$

where $\partial_i I_j$ denote the partial derivative of I_j with respect to x_i

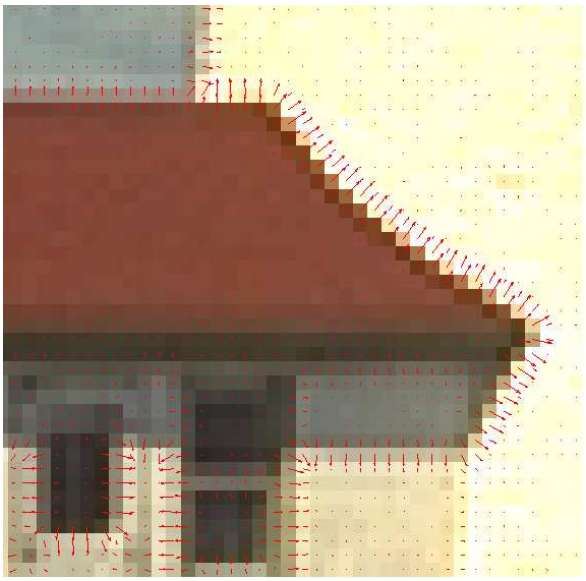


Figure 6.1 Field of normals of a color image

Field singularities

In general, a singularity is a point at which an equation, surface, etc., "blows up" or becomes degenerate. Singularities are often also called singular points. Singularities are extremely important in complex analysis, where they characterize the possible behaviours of analytic functions. We will give in the next section some elements to better understand the definition of a vector field singularity.

Singular points : A point $\mathbf{x}$ is a singular point of a vector field if $\eta(\mathbf{x}) = 0$. This singular point is said *simple* if the differential of the field is regular (have no zero eigenvalues). Moreover, the singular point is said *hyperbolic* if the differential of the field admits no purely imaginary eigenvalues. Furthermore, the Grobman-Hartman theorem shows that in the neighborhood of a simple and hyperbolic singular point, the field is topologically equivalent to its linear part provided that no eigenvalue of the linearization has its real part equal to zero. Thus, and considering the theorem says "local appearance", we are ensured that the topology of a sufficiently regular field is completely determined by its singular points. This means that the local structure of a regular field can be completely specified from the topological characterization of affine vector fields. We will apply this property to the vector field of normals to characterize the image topology.

Phase portrait : Let v be an affine vector field i.e. :

$$v(\mathbf{x}) = \mathscr{A}\mathbf{x} + \mathbf{b} \tag{6.3}$$

where $\mathbf{b}$ represents the translation and $\mathscr{A}$ the infinitesimal strain field tensor. If $\mathscr{A}$ is an invertible matrix , then the vector field has a unique singular point, given by:

$$\mathbf{x} = -\mathscr{A}^{-1}\mathbf{b} \tag{6.4}$$

which may not belong to vector field support (the support of a function is the set of points where the function is not zero-valued or, in the case of functions defined on a topological space, the closure of that set). The characterization of the field's structure is then fully determined by the spectral study of $\mathscr{A}$, which we call *phase portrait*. However for general image characterization the classification of such structures and their singular points is not necessary.

Detection algorithm

The detection algorithm is based on two key steps: the calculation of the vector field of normals and the research of singularities within the field. In the next section, we will detail each of these phases, both presented in the context of a multi-scale and multi-resolution scheme.

Computation of vector field of normals : As seen in equation 6.2, the computation of the vector field of normals go through the estimation of partial derivatives of the image. In order to do this, we choose to project the image components on polynomials of degree 1

of a Hermite basis $\mathbb{B} = B_{i,j}(\mathbf{x})_{\substack{i=0..d_1 \\ j=0..d_2}}$. As seen previously this bivariable polynomial basis is defined using the discrete inner product :

$$\langle F\,|G\rangle_{\omega} = \sum_{\mathbf{x}\in\Omega} F(\mathbf{x})G(\mathbf{x})\,\omega(\mathbf{x}) \tag{6.5}$$

with Ω the support domain of the basis. In this case we will use square supports of $(2L+1)\times(2L+1)$ sizes reported to $[-1;1]^2$ with respect to the associated values for $\mathbf{x}$ points of the discretization. Therefore the weighting function is then defined by:

$$\omega = exp(-\sqrt{L}(\sum_i x_i^2)) \tag{6.6}$$

that ensures multiscale approximation since the projection on a polynomial of this basis implies convolution with a Gaussian function. Next we define as *scale* the L parameter.

Figure 6.2 Test image (right) and restricted area

Thus the vector field of normals is given by :

$$\eta(\mathbf{x}) = \left(\sum_j \left\langle B_{1,0}\left|I_j^{\Omega|\mathbf{x}}\right.\right\rangle_{\omega}, \sum_j \left\langle B_{0,1}\left|I_j^{\Omega|\mathbf{x}}\right.\right\rangle_{\omega}\right) = (\eta_1,\eta_2) \tag{6.7}$$

where $I_j^{\Omega|\mathbf{x}}$ is the restriction to the subdomain Ω centered in the point $\mathbf{x}$ of the j^{th} component of the image. Furthermore the notion of multi resolution can be added by shifting the support of a D non-unitary value. Newt the parameter D will define the *resolutions*.

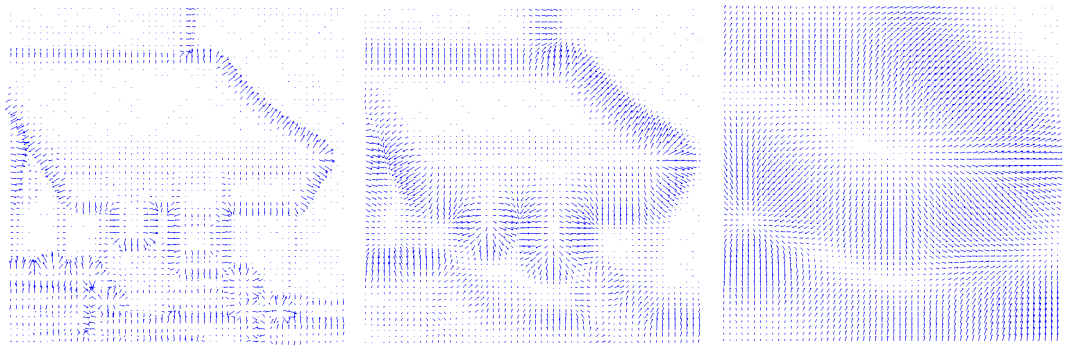


Figure 6.3 Identical vector field parts computed at different scales : left -*scale* = 1 et center - 4 and right 16 see restricted area in Figure 6.2.

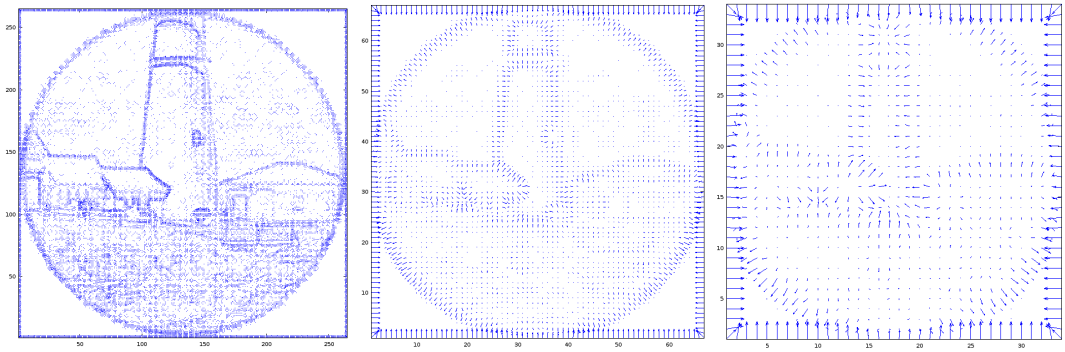


Figure 6.4 Vector fields computed at *scale* = 2 and *resolution* = 1 (left), *scale* =8 and *resolution* 4 (center), and *scale* = 16 and *resolution* 8 (d) see restricted area in Figure 6.2.

Furthermore, to normalize normal vector components between -1 and $+1$, the projections are divided by constants $C_{i,j}$ determined according to :

$$C_{i,j} = M \sup \left(\sum_{\Omega} B^{+}_{i,j}(\mathbf{x})\, \omega(\mathbf{x}),\ -\sum_{\Omega} B^{-}_{i,j}(\mathbf{x})\, \omega(\mathbf{x}) \right) \tag{6.8}$$

where M is the maximum coding value of the image and $B^{+}_{i,j}(\mathbf{x})$ (resp. $B^{-}_{i,j}(\mathbf{x})$) represent a positive (resp. negative) value of $B_{i,j}$ in the point x of the support.

6.2.2 Singularity search in a vector field of normals

We have seen previously that in the neighborhood of a simple hyperbolic singular point, the field is homeomorphic to its affine component. Therefore, to find the singularities of the field, we begin to extract the local affine components. For this we use again a polynomial bivariate

Hermite basis $\mathbb{B}^S = \left\{B^{\mathscr{S}}_{i,j}(\mathbf{x})\right\}_{\substack{i=0\cdots d_1\\ j=0\cdots d_2}}$ having $L_{\mathscr{S}}$ as scale and $D_{\mathscr{S}}$ as resolution features. So, the model of the affine vector field in equation 6.3 becomes an affine local approximation :

$$\tilde{\eta}(\mathbf{x}) = \begin{pmatrix} \left\langle \eta_1 | B^{\mathscr{S}}_{1,0} \right\rangle_{\omega^S} & \left\langle \eta_1 | B^{\mathscr{S}}_{0,1} \right\rangle_{\omega^S} \\ \left\langle \eta_2 | B^{\mathscr{S}}_{1,0} \right\rangle_{\omega^S} & \left\langle \eta_2 | B^{\mathscr{S}}_{0,1} \right\rangle_{\omega^S} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \left\langle \eta_1 | B^{\mathscr{S}}_{0,0} \right\rangle_{\omega^S} \\ \left\langle \eta_2 | B^{\mathscr{S}}_{0,0} \right\rangle_{\omega^S} \end{pmatrix} \tag{6.9}$$

and the search for possibly associated singular points is done by solving the equation 6.4, when $\mathscr{A}$ is invertible :

$$\mathbf{x}_s = - \begin{pmatrix} \left\langle \eta_1 | B^{\mathscr{S}}_{1,0} \right\rangle_{\omega^S} & \left\langle \eta_1 | B^{\mathscr{S}}_{0,1} \right\rangle_{\omega^S} \\ \left\langle \eta_2 | B^{\mathscr{S}}_{1,0} \right\rangle_{\omega^S} & \left\langle \eta_2 | B^{\mathscr{S}}_{0,1} \right\rangle_{\omega^S} \end{pmatrix}^{-1} \begin{pmatrix} \left\langle \eta_1 | B^{\mathscr{S}}_{0,0} \right\rangle_{\omega^S} \\ \left\langle \eta_2 | B^{\mathscr{S}}_{0,0} \right\rangle_{\omega^S} \end{pmatrix} \tag{6.10}$$

Still, for normalization, the projections are divided by the constant $C^{\mathscr{S}}_{i,j}$ computed as previously.

Thus, we consider singular points of an image the center points of $\Omega\,|_x$ subdomains where a singularity in the vector field of normals is detected. This applies when the affinity assumption of the vector field is approximately correct. Instead of measuring the validity of the affine model, we eliminate systematically singular points too distant from their support. Therefore only points satisfying the equation :

$$\|\mathbf{x}_s\|_1 < \delta_\Omega \tag{6.11}$$

where δ_Ω is a *restriction parameter* whose default value is set to 1 are considered as valid.

The matrix $\mathscr{A}$ is involved for "phase portrait" computation, and we can notice that it can be derived by projecting the vector field of normals on first order Hermite basis polynomials. Thereby, $\mathscr{A}$ can be seen as an approached form of the second order partial derivatives (sort of Hessian matrix) and consider therefore its eigenvalues as principal curvatures of the image. Particularly, the evaluation of the total curvature :

$$\lambda = \mathsf{tr}(\mathscr{A}) \tag{6.12}$$

and the gaussian curvature :

$$\gamma = \mathsf{det}(\mathscr{A}) \tag{6.13}$$

can be directly evaluated. The latter is directly involved in reversing the matrix $\mathscr{A}$. This information can be used when adopting the selection model of singular points, by keeping

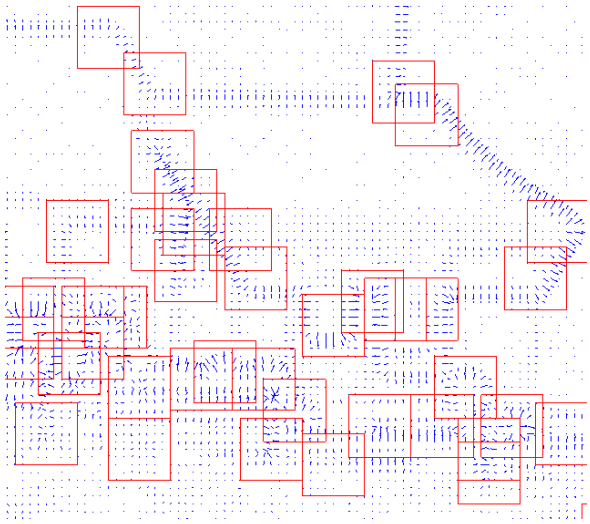


Figure 6.5 Extract of a normal vector field and its singular points detected using the next parameters :$L = 1$, $D = 1$, $L_{\mathscr{S}} = 2$, $D_{\mathscr{S}} = 1$, $\delta_\Omega = 1$, $L_V = 2$, $\delta_* = 0$. Red squares indicate the area in the selection of the singular point.

only dominant and robust points. We now consider the centered reduced distribution λ_* and γ_* of absolute values of λ and γ.

The robustness is ensured by a first decimation that eliminates any point for which λ_* is not a local maximum for a $(2L_V + 1) \times (2L_V + 1)$ neighborhood, where L_V is a *search parameter*.
The preponderance is ensured by a second decimation that eliminates every point that doesn't correspond to the notion of significant curvature, defined by :

$$\delta_* < \lambda_* < \gamma_* \tag{6.14}$$

where δ_* is a *search parameter* whose default value is 0.

6.2.3 Application of the raw method

The raw method involves applying a single pass of the process described above, namely a multi-scale and multi-resolution extraction of the normal field, and a multi-scale and multi-resolution detection of singularities of the vector field.

Configuration examples : The raw method is applied with three different configurations, having the same values for $\delta_* = 0$ and $\delta_\Omega = 1$ and the other next parameters :

- Local detection (C1): $L = 1$, $D = 1$, $L_{\mathscr{S}} = 1$, $D_{\mathscr{S}} = 1$ and $L_V = 1$
- Mean horizon detection (C2) : $L = 2$, $D = 1$, $L_{\mathscr{S}} = 2$, $D_{\mathscr{S}} = 1$ and $L_V = 4$
- Wide horizon detection (C3) : $L = 2^r$, $D = 2^r$, $L_{\mathscr{S}} = 2$, $D_{\mathscr{S}} = 2$ and $L_V = 4$

The r parameter in the third configuration is recursively determined by dividing by 2 the minimum dimension of the image, until this value is lower than 100.

The results are visually compared to those obtained with the two state of the art keypoint detection methods, namely SIFT and SURF. The latter are implemented using OpenCV library functions with default settings SIFT and the threshold for hessian keypoint detector used in SURF set to 800.

It should be noted that unlike our method that uses all the information of the RGB channels, these methods work using the "luminance" version of images.

Results : Raw method tests are presented in the 3×2 figures below Figures 6.6 , 6.7 and 6.8. The representation of singular points is approximate (singular areas are round instead of rectangular) to be comparable with the keypoint representation used in OpenCV.

Figure 6.6 Up : SIFT keypoint detection (left), Box image (input image), SURF keypoint detection. Down : Singular keypoint detection, using C1 (left), C2 (center) and C3 (right) parameters

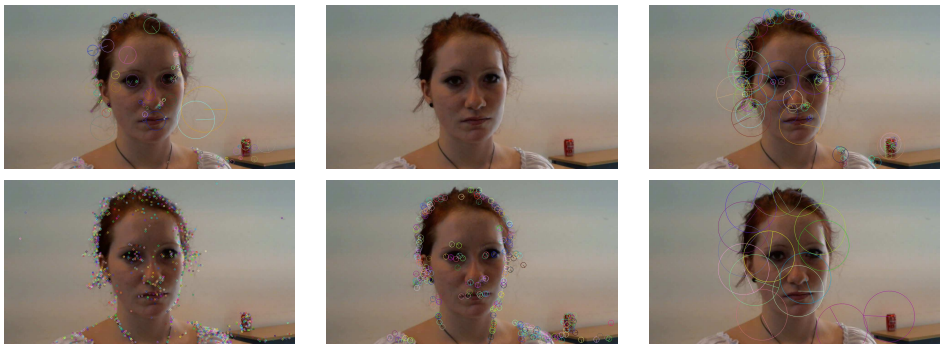


Figure 6.7 Up : SIFT keypoint detection (left), Girl1 image (input image), SURF keypoint detection. Down : Singular keypoint detection, using C1 (left), C2 (center) and C3 (right) parameters

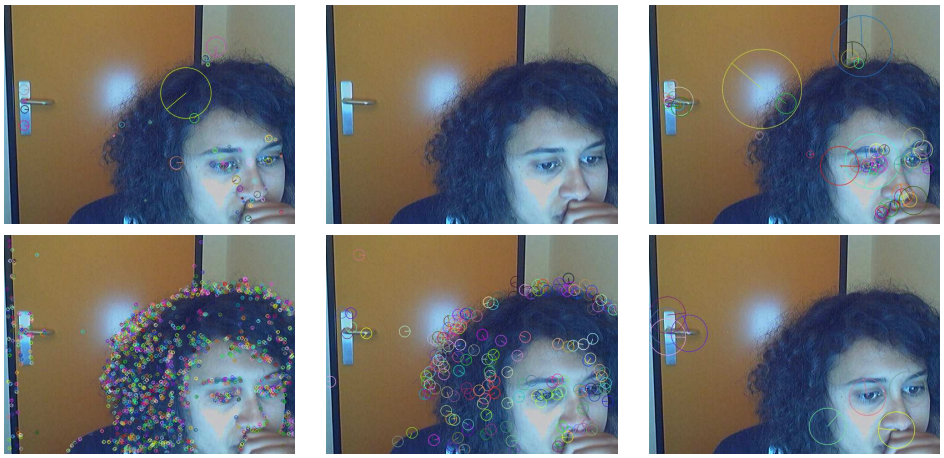


Figure 6.8 Up : SIFT keypoint detection(left), Girl2 image (input image), SURF keypoint detection. Down : Singular keypoint detection, using C1(left), C2(center) and C3(right) parameters

6.2.4 Selection strategy

We will show now that we can integrate into a general outline the detection research of singularities previously described. We will see the overall strategy allows to set the process parameters, in order to establish an evolution framework.

Pyramid representation

The pyramid representation is based on a first pyramid that specifies fields of normals supplemented for each level of resolution by a second pyramid of scales (see Figure 6.9). It is freely inspired by common strategies in the image processing field.

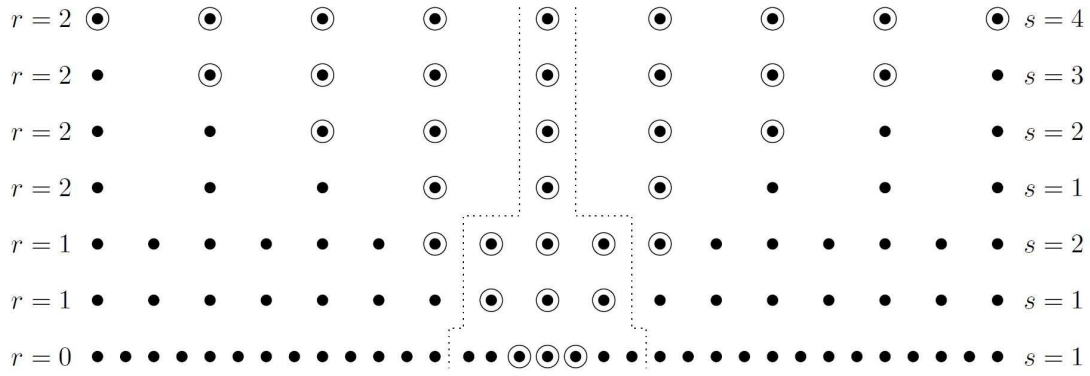


Figure 6.9 Pyramid representation using r (resolutions) and s (scales)

Fields of normals pyramid

The multi-resolution fields of normals pyramid is constructed jointly with a multi-resolution image pyramid. To build these pyramids, we must first consider the sizes $N1 \times N2$ of the images (resp. fields). Regarding to the image pyramid, it is a succession of avatars of the initial image $I^{(0)}$. The transition from one level (r) to the upper level $(r+1)$ is made by keeping the projections on the polynomial of degree 0 of a bivariate polynomial Hermite basis $\mathbb{B}^{\mathscr{R}} = B^{\mathscr{R}}_{i,j}(\mathbf{x})_{\substack{i=0..d_1 \\ j=0..d_2}}$ namely:

$$I^{(r+1)} = \left\langle I^{(r)} | B^{\mathscr{R}}_{0,0} \right\rangle_{\omega^{\mathscr{R}}} \tag{6.15}$$

This basis is configured using the parameter $L_{\mathscr{R}}$ for scale and $D_{\mathscr{R}}$ for resolution which, by default, are set to $L_{\mathscr{R}} = 1$ (for calculation speed) and $D_{\mathscr{R}} = 2$ (standard multi-resolution). The remaining parameter is the number of levels of the pyramid, that can be obtained by the number R such as :

$$N^{(R)} = \inf\left\{ r | N^{(r)} < N_{min} \right\} \tag{6.16}$$

using a constant N_{min} that we choose to be equal to 128. Thereby, to avoid the sub-resolution we supplement the previous rule with the next condition:

$$R = R - 1 \quad if \quad N^{(R)} < \frac{2}{3} N_{min} \tag{6.17}$$

The fields of normals pyramid is defined as the series of the fields $\eta^{\mathscr{R}}$ obtained according to Equation 6.7 applied to images $I^{(r)}$ using a basis that we denote $\mathbb{B}^{\eta}$. In order to ensure a

better disassociation between the influence of the various parameters and the overall research process, by default we use small scale and resolution characteristic $L_\eta = 1, D_\eta = 1$.

Scales pyramid

To deal with scale changes a scale selection method is applied. The idea is to select the characteristic scale of a local structure, for which a given function attains an extrema over scales. Singularities detection is based on the local assessment of an affine model (see Equation 6.10) which depends on a $\mathbb{B}^{\mathscr{S}}$ basis. For a given resolution level (r), we construct a scale pyramid by estimating the local affine models of the field $\eta^{(r)}$ using a sequence of $\mathbb{B}^{(s)}$ bases where (s) is the calculation scale. For the entire set of these bases the resolution parameter is constant, namely $D_{(s)} = 1$ in order to have a complete search. However, the scale parameter will vary in the range :

$$s \in 1...S^{(r)} \tag{6.18}$$

where the upper boundary $S^{(r)}$ depends of the resolution level. We use inherent basis properties when transforming by polynomial bases (see wavelet packet decomposition) to optimize the number of operations and avoid redundant calculations. These boundaries correspond to

$$S^{(r)} = (D_{\mathscr{R}})^r \tag{6.19}$$

and the bases $\mathbb{B}^{(s)}$ will have the scale parameter set to $L_{(s)} = s$.

In the previous section we have described some *local search parameters*. Their values have to be adapted to the new search algorithm. The parameter $L_{V,(s)}$ that describes the size of the neighborhood for extrema selection of the total curvature, is set equal to the scale s to reduce the number of parameters while maintaining a reasonable decimation

$$L_{V,(s)} = L_{(s)} = s \tag{6.20}$$

The parameter δ_* for significant curvatures selection (see Equation 6.14) remains constant for about the entire process. This choice is made to simplify the parameters of a comprehensive and global process. Finally, the parameter δ_Ω of singularity localization (see Equation 6.11) is also constant for the entire search process.

Global singularities selection

The detection process is, for now, designed to favour local representation and non-redundancy. It lacks, however, of a purely global stage to better guarantee that each singular point has its inter-resolution and inter-scale specificity. Let $\mathbb{X}$ be the set of singular points detected using the double pyramid system. Each point $\mathbf{x}$ has associated several characteristics :

$$\mathbb{X} = \mathbf{x} = ((\mathbf{x}^{(k)})_k, r, s, m, E, \theta) \tag{6.21}$$

where $(\mathbf{x}^{(k)})_k$ represent the set of points coordinates related to different resolutions, r the resolution level where the point was detected, s its scale, m the size characteristic , E the error and θ the associated orientation of the point. The size characteristic is calculated for the detected area of the point taken at the resolution 0 which size is $(2m+1) \times (2m+1)$ and therefore :

$$m = L_{(s)}(D_{\mathscr{R}})^{(r)} \tag{6.22}$$

where (r) and (s) are the scale and resolution at the detected scale. For the error characteristic we favour the points that are detected at a small resolution or at a bis scale, to ensure the robustness to smoothing. So,

$$E = m^{-1}||\mathbf{x}^{(r)}||_1 \tag{6.23}$$

The key point orientation is calculated from the field of normals at the resolution r where the singularity was detected using PCA (as PCA gives the direction along which data varies the most) on the key point size region.

The cleaning of $\mathbb{X}$ is the final step of our algorithm, where we eliminate each point that can suffer from side effects, keeping only the points that ensure :

$$m(x) < p_i^{(0)} < N_i^{(0)} - m(x) \tag{6.24}$$

where $N_i^{(0)}$ is one of the sizes of the initial image. The other cleaning strategy is related to the points density, we decide to keep only the points such as:

$$r(x) = \sup_{y \in U(x)} \{r(y)\} \tag{6.25}$$

where $U(x)$ is the $(2L_{U(x)}+1) \times (2L_{U(x)}+1)$ neighborhood of a point x. The size of the neighborhood can greatly influence the results. Two strategies are then possible - one with a big power of decimation where the size $U(x)$ evolves with the resolution level (for example,

$L_{U(x)} = R - r(x)$) and another strategy with a weaker decimation , where the size $U(x)$ stays fixed (for example, $L_{U(x)} = 1$)

After the cleaning of the entire $\mathbb{X}$ we can select the most relevant points with respect to their associated error. In order to do this, we choose the subset of $\mathbb{X}_a$ singular points such as:

$$\mathbb{X}_a = \mathbf{x} \in \mathbb{X} | E(\mathbf{x}) < \bar{E} - \alpha\sigma_E \tag{6.26}$$

where E is the $\mathbb{X}$ average error and σ_E the standard deviation of these errors. Again, it is obvious that the value of α can greatly influence the final number of selected points. However, the values of α situated around 1 seem to be sufficient in most cases where we primarily seek to restrict the number of singular points, the default value will be set to $\alpha = 1.15$.

Experimental results

Detection results obtained using the global algorithm are presented below, using the default parameters values namely : $L_{\mathscr{R}} = 1$, $D_{\mathscr{R}} = 2$, $L_\eta = 1$, $D_\eta = 1$, $L_{V,(s)} = L_{(s)} = s$, $D_{(s)} = 1$, $\delta_\lambda = 1$, $\delta_* = 0$, $L_{U,(x)} = 1$ and $\alpha = 1.15$

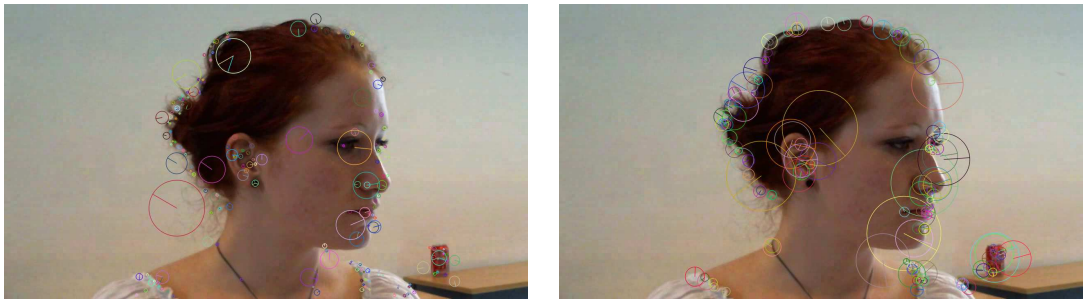


Figure 6.10 Left - SIFT detector (223) and right- SURF detector(144)

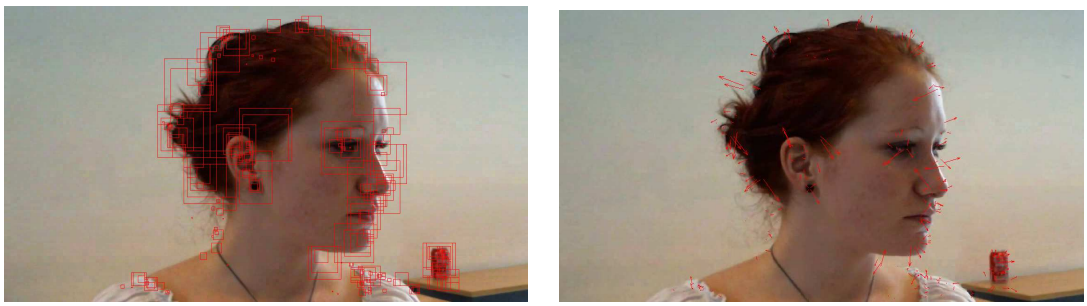


Figure 6.11 Polynomial sigularities detector sizes (left) and orientations (right)

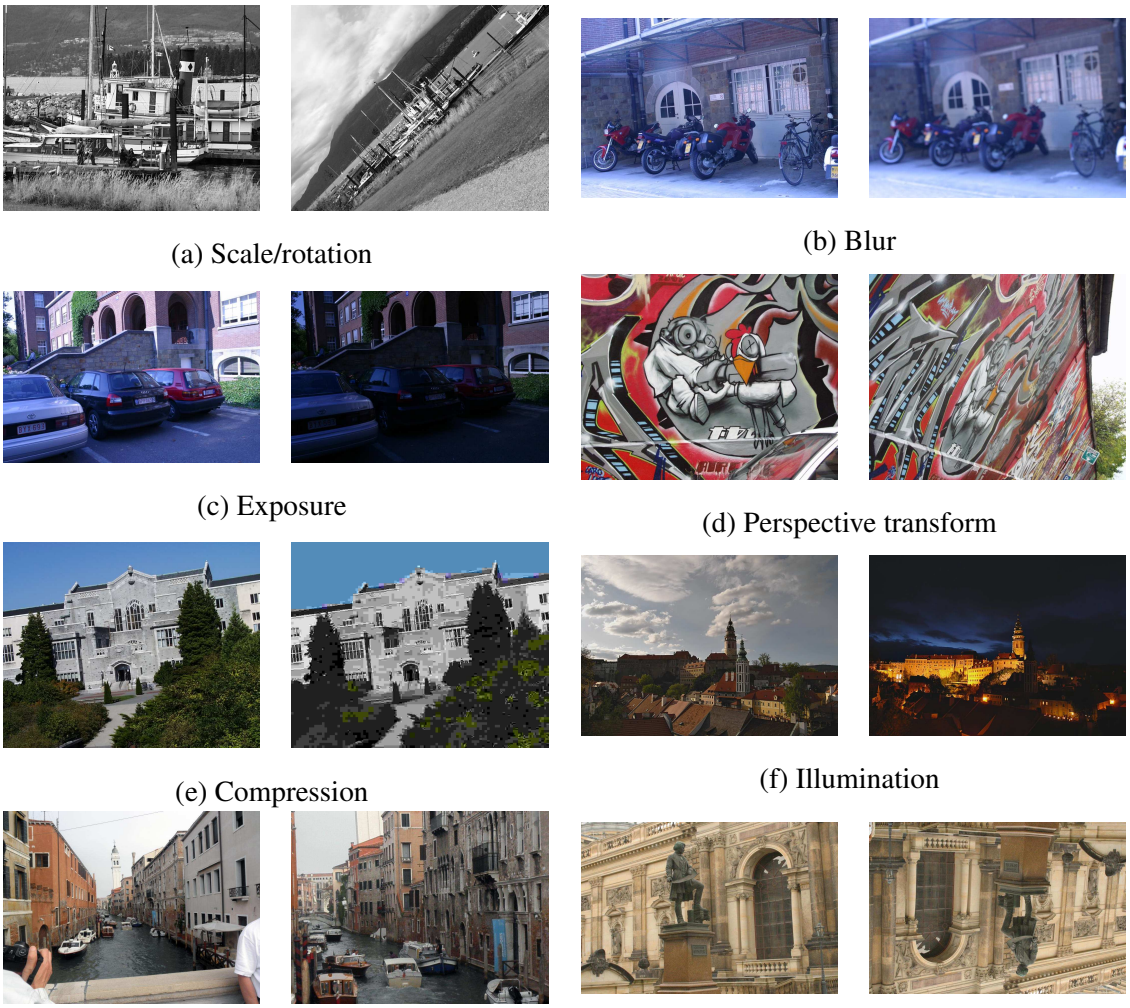
Figure 6.12 Left - SIFT detector (639) and right- SURF detector(1127)

Figure 6.13 Polynomial sigularities detector (524) sizes (left) and orientations (right)

6.3 Evaluation on Oxford dataset

In 2005, Mikolajczyk et al.[MS05] evaluated affine covariant region detectors, and compared their performance on a set of test images under varying imaging conditions. They have defined some performance measures for detector evaluation and also provided a dataset of benchmark image sequences (the Oxford dataset) to test the effects of blur, compression, exposure, scale/rotation, and perspective change. Each image sequence is composed by six images with a gradual geometric or photometric transformation and the ground truth homographies between the first image and the rest of images in the sequence.

In the perspective change test the camera varies from a fronto-parallel view to one with significant foreshortening. The scale change and blur sequences are acquired by varying the camera zoom and focus respectively. at approximately 60 degrees to the camera. The scale changes by about a factor of four. The light changes are introduced by varying the camera aperture. The JPEG sequence is generated using a standard image browser with the image quality parameter varying from 40 to 2%.



(a) Scale/rotation

(b) Blur

(c) Exposure

(d) Perspective transform

(e) Compression

(f) Illumination

(g) Zoom

(h) Rotation

Figure 6.14 Example images used for the evaluation tests. Images (a) - (e) are from the Oxford dataset, Images (f)-(h) are from the additional dataset

In addition to the 8 sequences of images from the Oxford dataset we use a supplementary dataset of 5 sequences from the dataset of Jared Heinly [HDF12] - 2 sequences that have a rotation transform between 0 and 180°,one sequence with a rotation transform between 0 and 360°, one sequence with a zoom transformation and one with an illumination change. Some example images used in our tests are presented in Figure 6.14.

In addition to our detector, we analyse the performance characteristics of nine recent key point detection methods : AKAZE [ANB13], BRISK [LCS11], FAST [RD06], KAZE [ABD12], MSER [DB06], ORB [RRKB11], SIFT [Low99], STAR [AKB08] and SURF [BTVG06].

Keypoint detectors

In this subsection we give a short description of the recent keypoint detectors.

- **SIFT** algorithm uses a set of sub-octave Difference of Gaussian filters, looking for 3D extremas in the resulting structure, and then computing a sub-pixel space and scale location using a quadratic fit. To achieve invariance to image rotation each keypoint is assigned one or more orientations. The SIFT detector is invariant to translation, rotations, and re scaling of the image.

- **SURF** detector proposes an efficient computation of features similar to SIFT, where, the Hessian matrix is approximated and gradients are calculated by a set of box-type filters and integral images.

- **FAST** compares pixels on a ring centered at a feature point. The algorithm works in two steps: in the first step, a segment test based on the relative brightness is applied to each pixel of the processed image, followed by an refinement step that allows to narrow the results using non-maximum suppression. FAST is several times faster than other existing corner detectors but it is not robust to high levels of noise.

- **MSER** algorithm extracts from an image a number of co-variant regions, called Maximally Stable Extremal Regions. An MSER is a stable connected component of some level sets of the image. Optionally, elliptical frames are attached to the MSERs by fitting ellipses to the regions.

- **STAR** also known as CenSurE detector computes the extrema of the center-surround filters over multiple scales, using the original image resolution for each scale. They are an approximation to the scale-space Laplacian of Gaussian and can be computed in real time using integral images.

- **BRISK** points of interest are identified across both the image and scale dimensions using a saliency criterion. In order to boost efficiency of computation, keypoints are detected in octave layers of the image pyramid as well as in layers in-between. The location and the scale of each keypoint are obtained in the continuous domain via quadratic function fitting.

- **ORB** uses Fast multi-scale detection with an efficiently computed corner orientation using the intensity centroid method.

- **KAZE** approach uses a nonlinear diffusion (variable conductance diffusion) to detect keypoints in nonlinear scale spaces keeping important image details and removing

noise. The nonlinear scale space is build efficiently by means of Additive Operator Splitting (AOS) schemes.

- **AKAZE** uses a novel mathematical framework called Fast Explicit Diffusion (FED) embedded in a pyramidal framework to speed-up dramatically the nonlinear scale space computation.

Performance Metrics

All the images of each sequence are related by a 2D homography H. This known transformation is used as ground truth data, allowing to know where a point p_A, extracted from an image A, should be projected in image B of the same dataset by $p_B = Hp_A$. Similarly, points extracted from the image B can be projected back to image A by using the inverse of H.

We first use the *repeatability* measure describing how well the detectors determine corresponding scene regions. This is measured by comparing the overlap between the ground truth and detected regions, using the overlap error , defined as :

$$\varepsilon_S = 1 - (\mathbf{A} \cap \mathbf{H^T B H}) / (\mathbf{A} \cup \mathbf{H^T B H}) \tag{6.27}$$

where **A** and **B** are the regions and **H** is the homography between the images. A match is assumed correct if the error in the image area covered by two corresponding regions is less than 50 percent of the region union, that is, $\varepsilon_S < 0:5$.

To determine the correspondent of a match, we use ground truth data to warp keypoints from the first image of the dataset into all remaining images. The *repeatability* score for a given pair of images is computed as the ratio between the number of correct matches and the number of regions in the train images.

Knowing that in a practical application regions need to be matched or clustered, and apart from the repeatability of the detection the distinctiveness of the region is important, we then compute a descriptor for the regions and then check to what extent matching with the descriptor gives the correct region match. In our tests we use the Latch descriptor [LH15] that won the CVPR 2015, OpenCV State of the Art Vision Challenge, in the Image Registration category.

To compute matches we adopt a test that compares the ratio of distances between the two best matches for a given keypoint, and rejects the match if the ratio is above a threshold of 0.8 for all tests (introduced in [Low04]).

The ratio of correct matches for each descriptor to the total number of ground truth matches is defined to be the *precision*.

Finally we also measure the *computation time* per frame and per keypoint. They have all been measured on a Intel Xeon, 2.53 GHz, Windows PC.

6.3.1 Feature extraction time comparison

In this section the extraction times per frame, quantities of keypoints and extracted time per keypoint are compared for all the detectors described above. All results are computed on the entire set of images from the Oxford dataset and the additional sequences 10 times and the mean results are given in the figure 6.15 .

Detector	Keypoints	Time per frame(ms)	Time per Keypoint(ms)
AKAZE	2900.6	245.473	0.109424
BRISK	6826.06	126.4728	0.020359
FAST	14304.6	7.815764	0.000738
KAZE	2813.66	746.2391	0.356434
MSER	927.489	572.4679	0.721546
ORB	498.068	24.6045	0.049343
POLY(ours)	3141.97	594.283	0.216903
SIFT	5478.33	340.1102	0.101359
STAR	889.148	37.58656	0.075767
SURF	5324.18	166.3272	0.034757

Figure 6.15 Averaged computation times for the different detectors

We compare our method to OpenCV implementations of the other detectors that are efficiently optimized. In comparison to other methods we can see that the Poly method is not very efficient. Presently our method outperforms MSER and KAZE keypoint detectors. However,we think that it is still possible to optimize our method in order to be fully comparable with the other keypoints detectors.

The largest number of keypoints are extracted by the FAST detector and the least number of keypoints is provided by ORB (due to the default OpenCV parameters). The variation in the number of features is expected, since the various detectors respond to different types of image structures. The most performant feature detector is FAST which is 48 times faster than SIFT and 95x faster than KAZE. On average FAST takes 7ms per image and our method 595ms.

6.3.2 Results under controlled transformations

To verify the accuracy of the different detectors to a separate transformation , we deform our image using the next transformations: rotation, scaling, blur and illumination. All the tests work in an analogous way: using a given source image (the first image of each sequence of our database) synthetic data is generated using the known transformation. Depending on the test case, we use the following algorithms:

- Rotation - rotation of the image source image around the Oz axis in degres (from 0 to 360).
- Scale - resizing of the source image from $0.25X$ to $2.0X$ of the original size
- Illumination - changing the overall brightness of the source image(from -127 to 127)
- Blur - adding of a gaussian blur (the kernel size varies from 3 to 17 pixels)

All transformations are performed on the first image of the 13 sequences from the image dataset.

Figures 6.16, 6.17,6.18,and 6.19 show the comparison of all the detector performance under rotation, blur, brightness, and scale transformations, respectively.

It can be observed that for rotation transformations our method performs similarly (the shape of the curve) to SURF and STAR detectors, ie the repeatability score i more sensitive to orientations like 45, 125 and 225 degrees, but with lower repeatability scores.

As for the precision, it can be observed that the detectors can be divided in two categories, those who have a good precision ie the number of correct matches is close to the number of all matches such as SIFT, ORB or SURF and those which are not robust against rotations larger than 30 degrees , such as KAZE, STAR or AKAZE. The average precision given by our detector is 50%, and it can be concluded that our method, even if we give an orientation for each keypoint is not robust to rotation changes for the moment and we should change our method of orientation assignment. FAST detector is not designed to be robust to rotation changes therefore it gives the worse results.

With respect to blur transformation, our method is the one that gives the best results for the most important filter size. The poorer results for this type of transform are obtained using Mser, SIFT and ORB detector. Blur changes may affect the keypoint's scale and orientation calculation, leading to a worse precision. Yet, it can be seen that ORB and SIFT have good precision results, due to the good performance of the Latch feature descriptor that overcomes this errors.

Figure 6.16 Repeatability (up) and precision (down) under rotation transformations

For brightness transform it can be observed that our detector gives slightly inferior results in terms of repeatability. As for the precision results, they are very similar to the other methods. Mser detector appears to be the most sensitive to this kind of transformation.

Since the FAST detector is not invariant to scale transformations, we did not showed it's results in figure 6.19. The best repeatability rates are observed for AKAZE. It is interesting to note that ORB has a good repeatability rate for large scales, and a poor one for small scales, which is the inverse for SURF. The results of our detector are similar to STAR, SIFT and BRISK and they outperform KAZE for most of the scale factors and MSER. In terms of precision, our method is the least performant, we suspect that mixing our detector with a Latch descriptor is not very efficient.

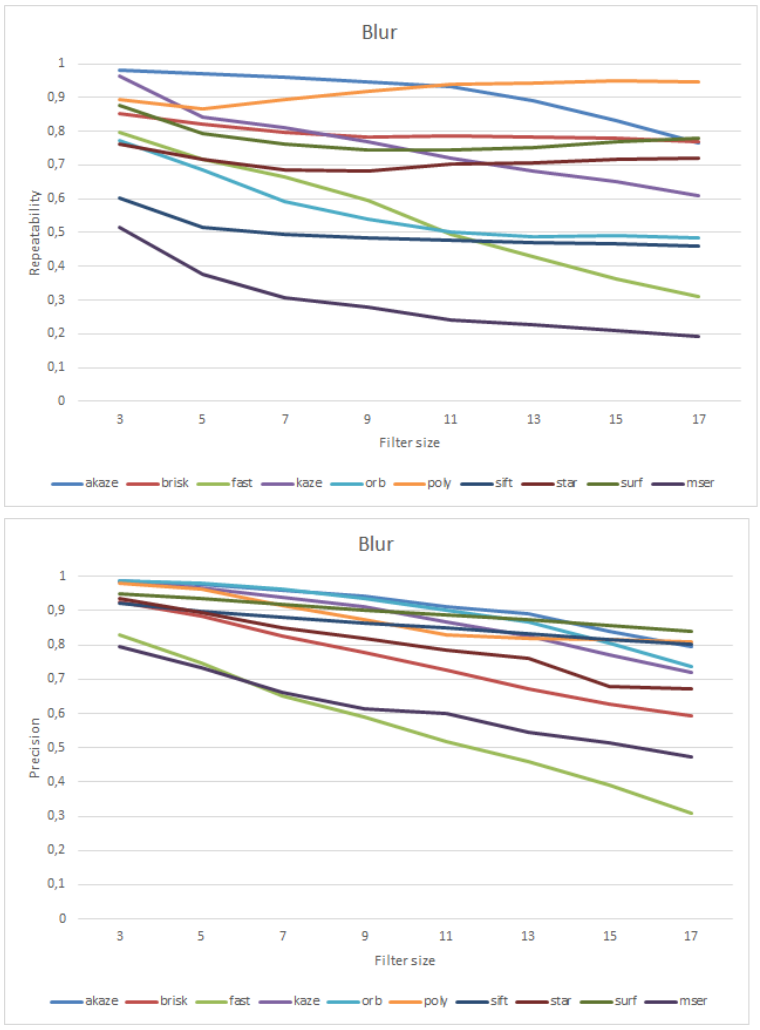


Figure 6.17 Repeatability (up) and precision (down) under blur transformations

6.3.3 Matching experiments

We now display real matching situations, using all images form the dataset joined with the homography matrices. Only transformations that were not presented overhead will be discussed.

For each transform the left side of the results present small transforms while the right side corresponds to larger transform, resulting in low quality images and/or in images where objects are very different from the ones in the reference image and therefore difficult for matching.

Figure 6.20 shows the mean results for light changes for the images on Fig.6.14(c) and (f). Compared to the other detectors our method presents very good repeatability results, especially for large transforms. However using the combination detector /descriptor the

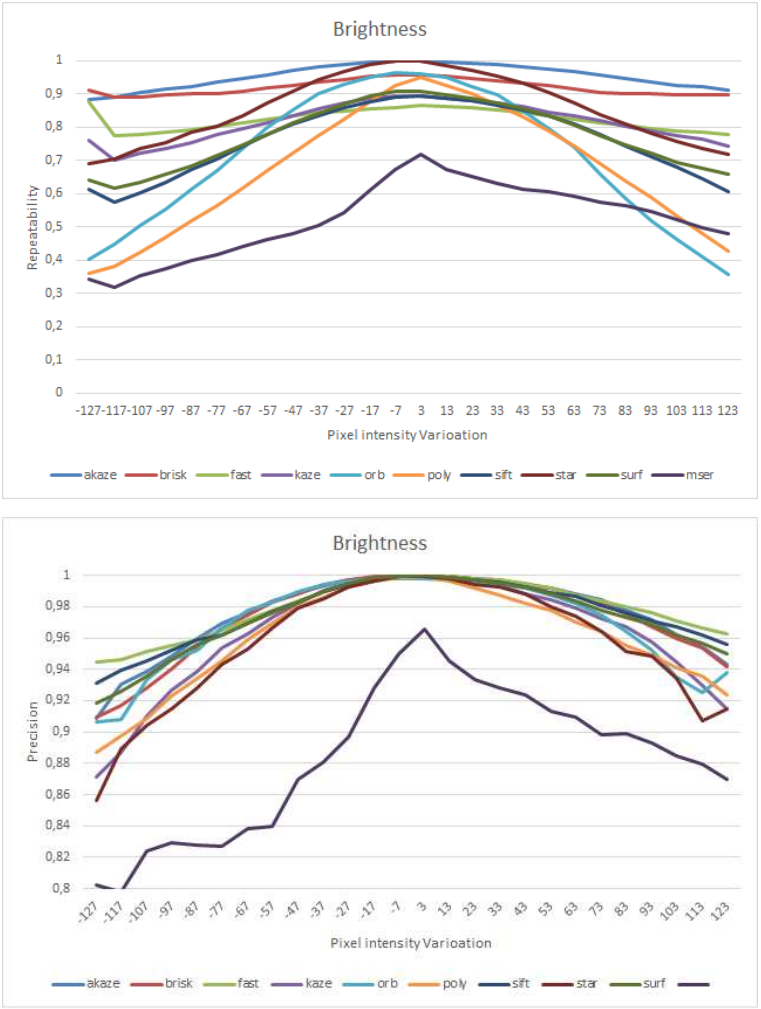


Figure 6.18 Repeatability (up) and precision (down) under brightness transformations

precision results are quite low. All the other curves show good robustness to illumination changes, having good repeatability and precision scores.

Figure 6.21 shows the score for the JPEG compression sequence from Fig 6.14 (e). For this type of transform the detectors having the higher repeatability rate are KAZE an AKAZE. Keypoint detection in the nonlinear space seems to be the more adapted to image compression. While constructing the scale-space pyramid the Gaussian blurring does not respect the natural boundaries of objects and smoothes in the same degree details and noise when evolving the original image through the scale space. As the noise becomes very important for high compression rates the use of non linear diffusion allows keeping important image details and removing noise. The degradation under increasing compression artefacts is similar for all

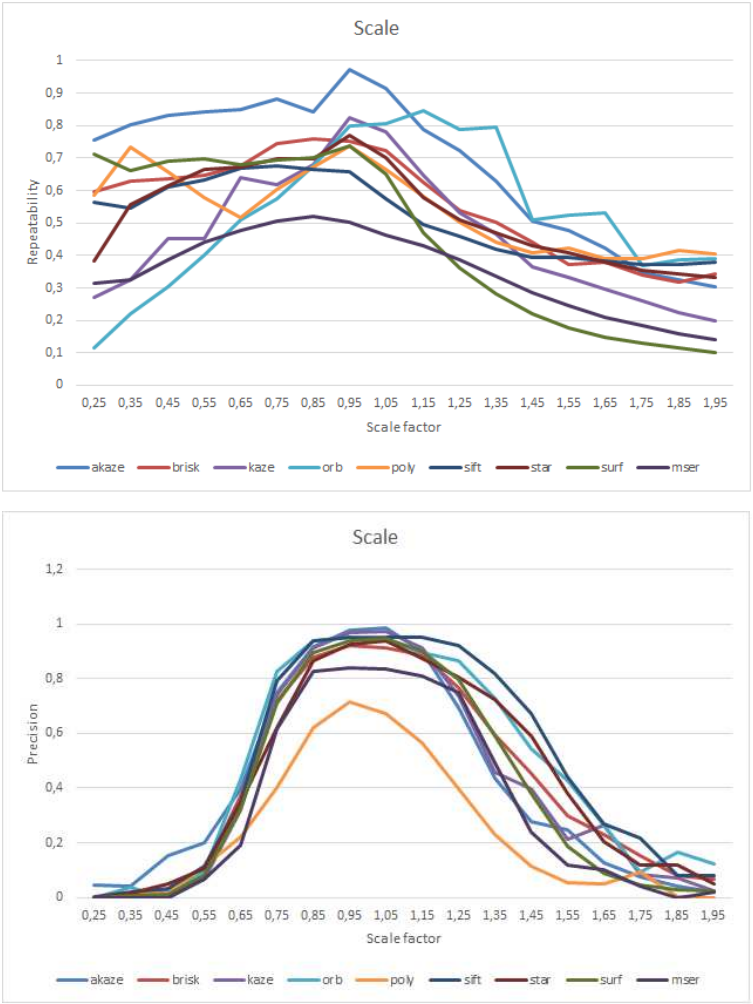


Figure 6.19 Repeatability (up) and precision (down) under scale transformations

detectors. Although our method presents lower repeatability results than AKAZE and KAZE, it outperforms most of the other keypoint detectors.

Figure 6.22 shows the mean results for the boat sequence from Fig 6.14 (a), and the bark sequence . The main image transformation is a scale change and in-plane rotation. In terms of repeatability the ORB keypoint detector performs best, followed by SIFT and our Polynomial detectors. The table 6.1 shows the mean repeatability scores of all the detection methods for each image from the two sequences with zoom and rotation transformations. We notice a considerable degradation of the repeatability and precision scores between the images boat, that have a structured scene and the bark sequence having a textured scene (presenting some leafs). The scene type is therefore very important for the selection of the keypoint detector.

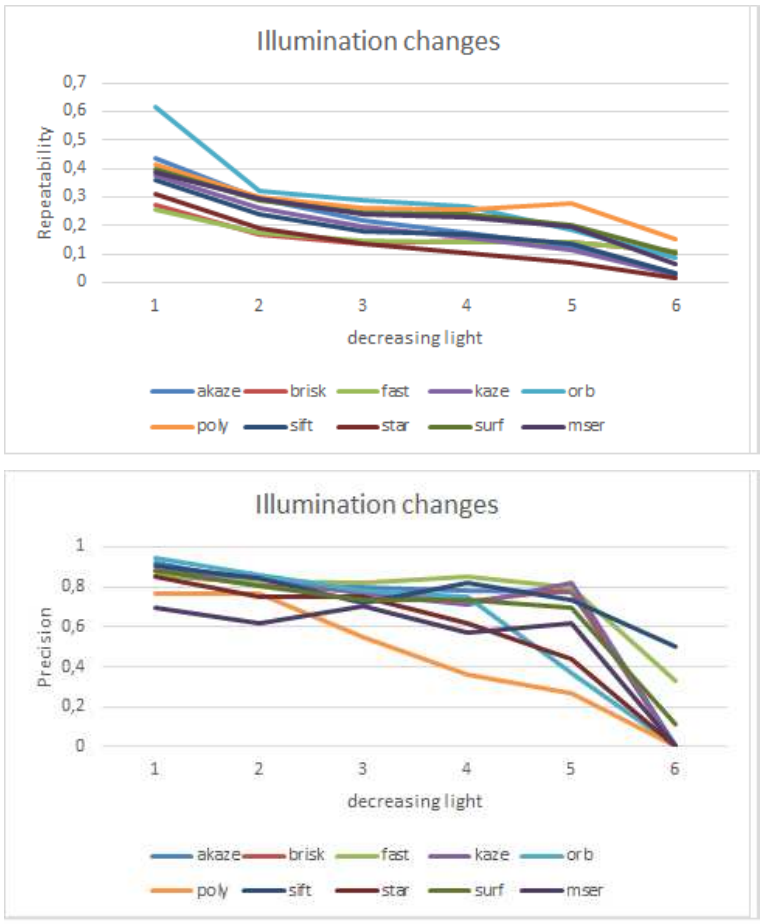


Figure 6.20 Repeatability (up) and precision (down) under illumination changes

Figure 6.23 presents the comparison results under a perspective transform. It can be observed that for small viewpoint changes the best detectors are ORB, MSER and AKAZE in terms of repeatability and SIFT, ORB and BRISK in terms of precision. The average results of our detector are explained by its sensitivity to rotation transforms. The repeatability score for a viewpoint change of 20 degrees varies between 18%and 52% and decreases for large perspective transforms to $4\% - 11\%$.

6.4 Regions of interest in a face alignment algorithm

During the evaluation stage of our keypoint detector we have noticed that in comparison to the sift or surf detectors on face images the areas that were selected with our method where more significant ie our detector always has chosen the eyes, nose and mouth areas and ignored regions with constant texture like the cheek or front skin. Since in the search

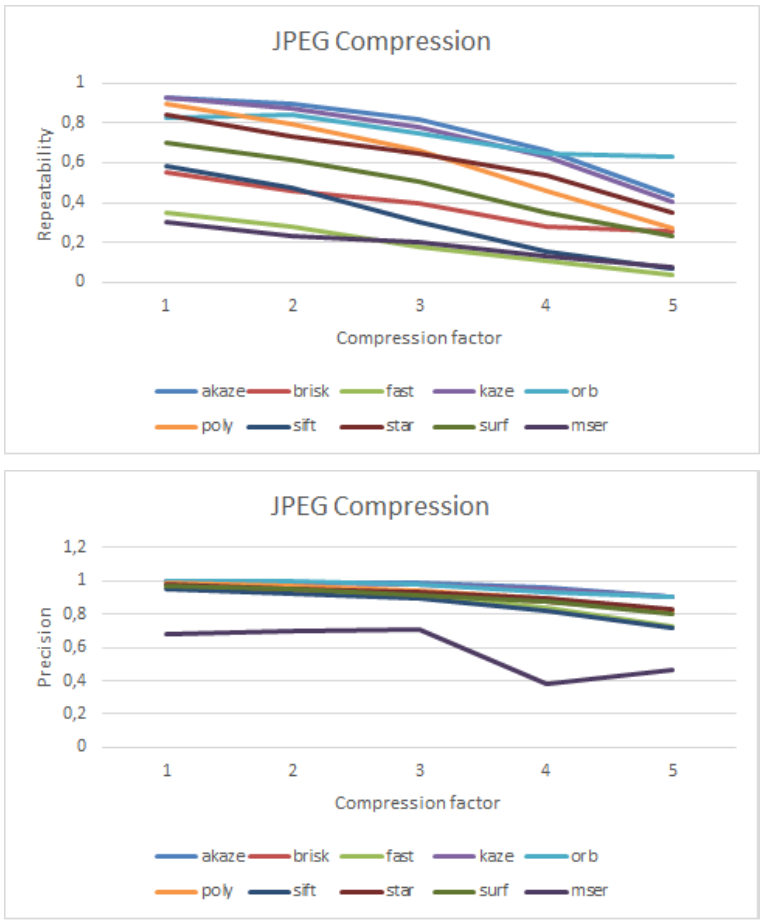


Figure 6.21 JPEG compression repeatability (up) and precision scores(down)

process we keep only dominant and robust points, in face images such points correspond to key regions we decided to test an alignment algorithm.

We decided to implement an algorithm using a cascaded regression similar to the one used in the compressed discriminative AAM (Algorithms 3 and 4) that uses interest key areas as features for the regression algorithm. First we train an algorithm that uses the warped texture inside the shape as feature in the regression.

The evaluation stage is similar to the one used in Algorithm 4. The main difference between the two algorithms is that in this chapter we use only a shape model in the training stage, and not an combined shape and appearance model used in Algorithm 4.

Then we modify the Algorithms 8 and 9 in order to integrate key areas regions as features in the cascaded regression instead of the whole face texture. In addition to the regions detected with our polynomial method we evaluate the ones found with the SIFT and SURF approaches. Therefore step 4 of the 8 and step 3 in the 9 are changed such as $\mathbf{f}_i =$ correspond to the concatenation of the detected key regions in the face images.

Algorithm 8 Training of cascaded regression

Input: N images with landmarks annotations

1: Build a statistical shape model
2: **for** $t = 1$ to T **do**
3: $\left\{\delta q_{p,t}\right\}_{j=1}^{S}$ Sample perturbations
4: $\mathbf{f}_i$ = the set of pixel intensities within the convex hull of the shape given by $\delta q_{p,t}$
5: **if** t>1 **then**
6: Estimate $(\delta q_{p,t})$ using R_{t-1} and add it to the current parameters
7: **end if**
8: $(\tilde{q}_{p,t}) = (q_{p,t}) - (\delta q_{p,t})$
9: $R_t = \arg\min_R \sum_i d(R(\mathbf{f}_i), (\tilde{q}_{p,t}))$
10: **end for**
11: Output $R = (R^1, ..., R^T)$

Algorithm 9 Evaluation of the cascaded regression

Input: Image I, initial shape S^0 and the set of regressors $R = (R^1, ..., R^T)$

1: Project shape to $q_{p,t}$ parameters
2: **for** $t = 1$ to T **do**
3: $\mathbf{f}_i$ = the warped texture inside the shape given by $q_{p,t}$
4: $(\delta q_{p,t}) = R_t(\mathbf{f}_i)$ // evaluate regressor
5: $(q_{p,t}) = (q_{p,t-1}) + (\delta q_{p,t})$ // update $(q_{p,t})$ parameters
6: Back project parameters to compute S^t
7: **end for**
8: return fitted shape

Images	Repeatability		Precision	
	Boat	Bark	Boat	Bark
1	0.4117	0.1587	0.6598	0.3765
2	0.2294	0.0455	0.2086	0
3	0.0821	0.0352	0	0
4	0.05092	0.0265	0	0
5	0.0436	0.0177	0	0

Table 6.1 Mean repeatability scores for Bark and Boat sequences

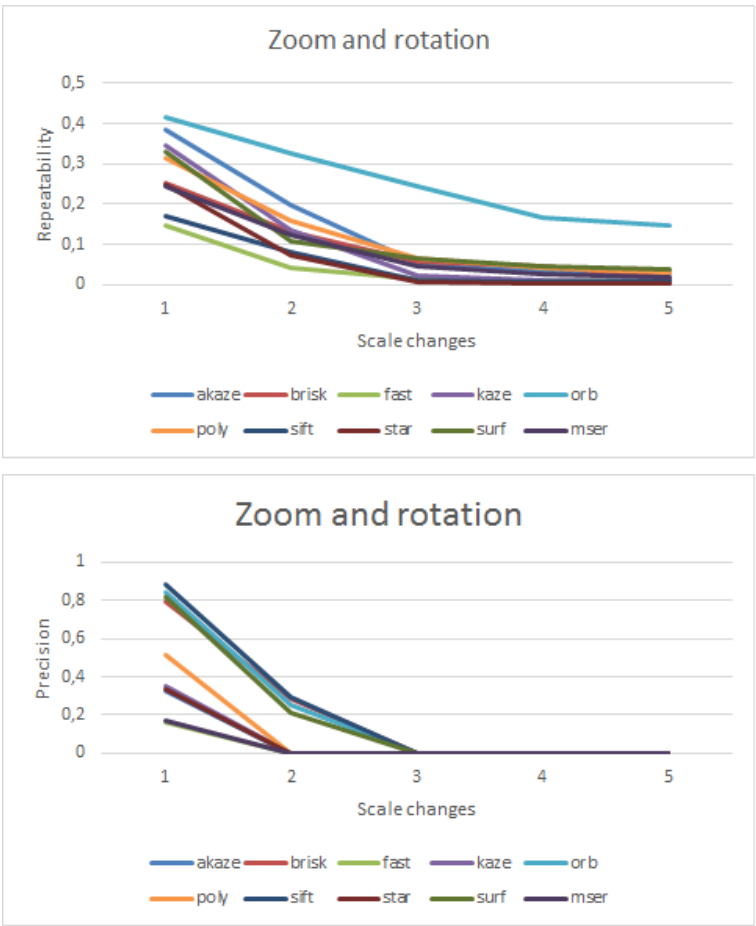


Figure 6.22 Repeatability (up) and precision (down) under zoom and rotation changes

An example of the features used in our algorithms for three different images is given in Figure 6.24.

We may observe that in the regions detected by SIFT there is always a big centered one that could be used to detect the face in several images. SURF also selects big regions around

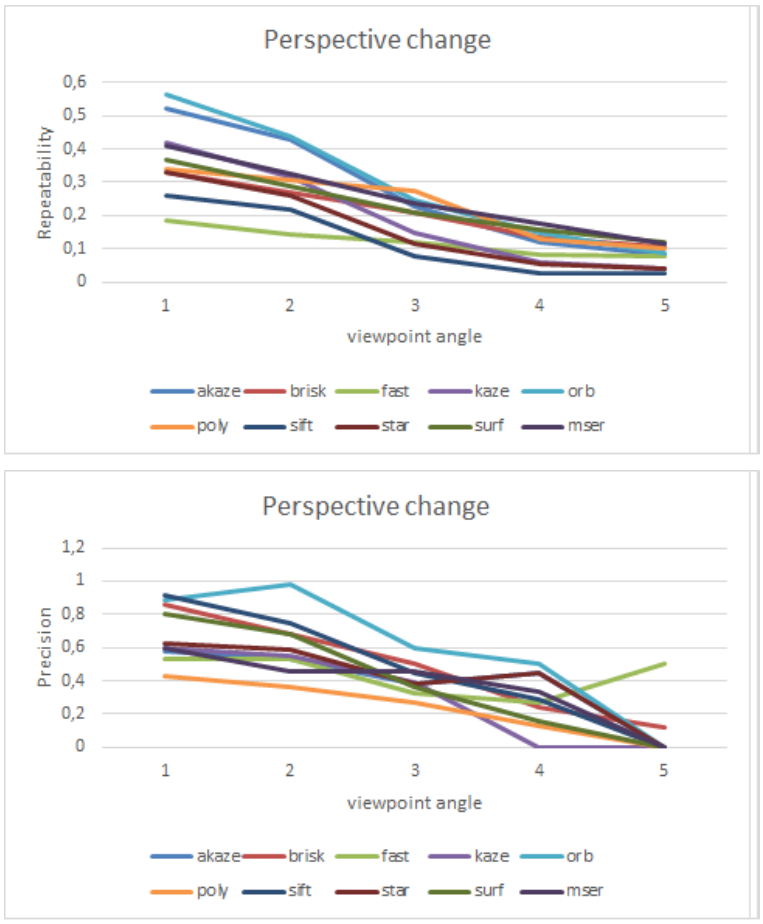


Figure 6.23 Repeatability (up) and precision (down) under zoom and rotation changes

the eyes, the nose and the lips. Our method gives smaller and more accurate regions of interest.

6.4.1 Experimental results

For each database we have trained four different models - one using the entire texture and three that employ face regions of interest detected by SIFT, SURF and our polynomial method. In the algorithm using SURF, as previously we have set the Hessian threshold to 800.

Table 6.2 shows the comparative fitting results on the MUG database. It can ve observed that the SIFT, SURF and the method using the entire texture present similar results, and the approach using polynomials outperforms them. The number of interest regions is similar between SIFT and Polynomial detectors, and SURF detects much more regions (also visible in Figure 6.24).

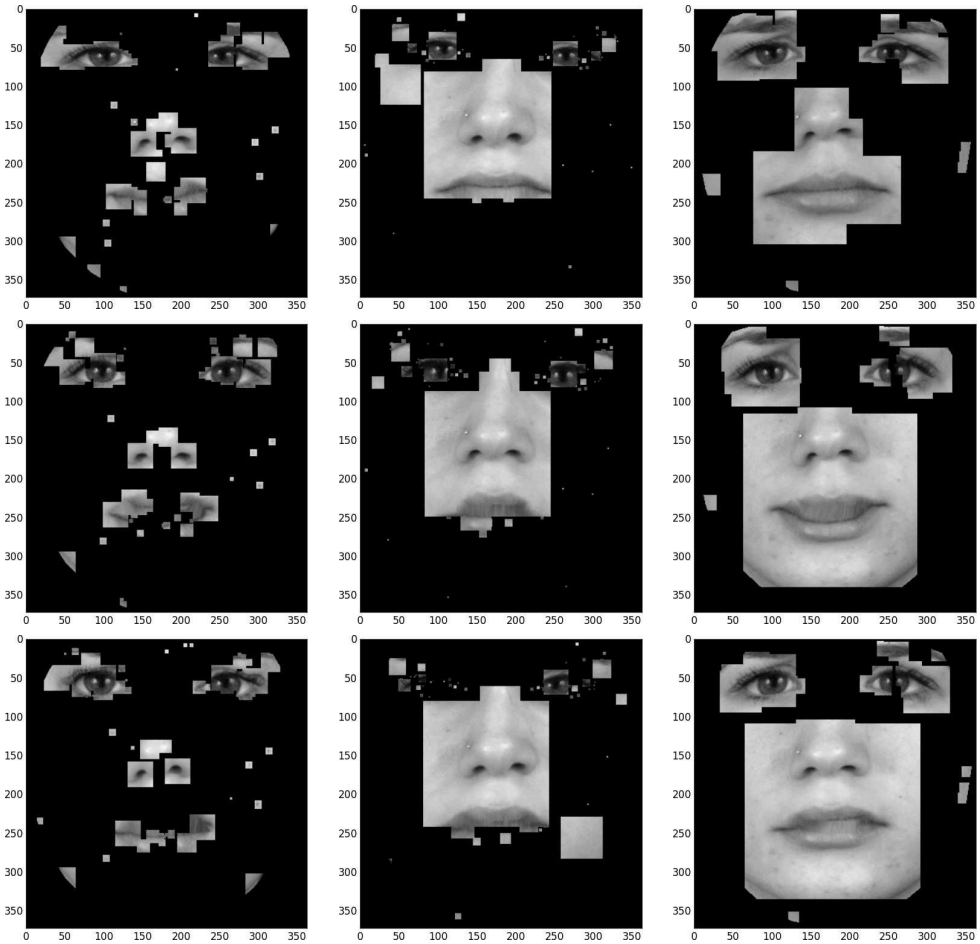


Figure 6.24 Detected areas in a face image

	Mean Error	Mean error on face interior points	Mean number of key regions
Raw	0.0791	0.0693	
Poly	**0.0720**	**0.0613**	100
SIFT	0.0793	0.0678	104
SURF	0.0787	0.0675	275

Table 6.2 Experimental results on MUG database

Table 6.3 shows the comparative results on IMM, MultiPIe and Cohn-Kanade databases where ME means the mean error and ME IP the mean error on face interior points. We can observe that for the 3 databases the best method is the one using the entire texture inside the convex hull of the shape. Since the dimensions of faces are quite small in this three databases, it is preferred to use all the information, and not only in the regions of interest, insufficient

Methods	IMM		Cohn-Kanade		MultiPie	
	ME	ME IP	ME	ME IP	ME	ME IP
Raw	**0.1049**	**0.0797**	**0.0690**	**0.0594**	**0.1094**	**0.0970**
Poly	0.4027	0.3489	0.1873	0.1692	0.2917	0.2589
SIFT	0.2892	0.2530	0.1677	0.1558	0.2469	0.2284
SURF	0.3086	0.2631	0.1414	0.1216	0.2386	0.2122

Table 6.3 Experimental results on IMM, Cohn-Kanade and MultiPie databases.

for a good alignment. It can be also observed that for IMM database the regions detected with our method are not pertinent to the algorithm. As we showed before, our method is sensitive to rotations, and since in IMM there is an orientation variation our approach is not suited for that case.

6.4.2 Discussion

We have presented in the first part of this chapter an original method for interest keypoint/zones detection. The presented approach works on color and grayscale images, the number of features is adaptable over a large range by a simple threshold and gives the possibility to use different types of bases (and therefore use different types of smoothing in the construction of scales pyramids) while creating the multivariate basis. Furthermore our approach is robust to scale, illumination and blur changes. Even if we give an orientation for each keypoint our keypoint detector is sensitive to rotations, therefore we should change our method of orientation assignment.

In the second part of the chapter we have used the detected regions instead of the entire texture as feature in a regression algorithm. Results show that for databases comporting facial rotations our approach is not relevant and that for databases where the face is of reasonable size(superior that 300 pixels such in MUG database) our approach gives better results than using the entire texture information.

In the next chapter we will detail how the results of a tracking algorithm can be exploited to the description of expression and present a polynomial based texture representation model as a descriptor for facial expression information.

Chapter 7

Polynomial based texture representation for facial expression recognition

7.1 Introduction

In this chapter, we propose a new polynomial based texture representation method for extracting information about facial expressions. While many appearance-based methods have been proposed over the years to improve the performance of facial expression recognition, most descriptors are usually unable to both provide precise multi-scale / multi-orientation analysis and handle the redundancy problem effectively.

The automatic recognition of facial expressions is one of the most challenging and popular topics in the computer vision domain as it impacts important applications such as virtual reality, broadcasting, user profiling or video conferencing.

An essential step for a successful facial expression recognition is the extraction of facial features that attempt to find the most effective representation of face images. There are two common feature extraction approaches : geometric feature-based systems, using major face components and/or feature points, and appearance based systems using image filters. A thorough survey of the existing work can be found in [WFY12, Bet12, SGM09]. Experimental results show that methods using Gabor wavelet transforms, derived from biological principles on the visual system, provide superior performance and are an effective method for facial expression recognition [ZLSA98, BLF+05]. However, it is both time and memory intensive to convolve face images with a bank of Gabor filters to extract multi-scale and multi-orientation coefficients.

In this chapter, we investigate the use of coefficients resulting from polynomial projections for texture representation within a system of facial expression recognition. We show in section

7.2 our proposed method and we compare in term of computational efficiency polynomial transforms to Gabor transforms. Experimental results obtained by applying the proposed technique on MUG [APD10a] and the extended Cohn-Kanade [LCK+10] databases are provided in section 7.3 and show significant recognition rates over state-of-the-art methods. Finally, we will conclude and open the discussion on further works in section 7.4.

7.2 Base projections for facial expression recognition

The orthonormal polynomial decomposition allows to extract the different frequency components of a signal and offers the possibility to use multiresolution piecewise polynomial decomposition, so it can be used for the feature extraction within a system of facial expression recognition.

We use as input to our approach still face images labeled with landmarks around fiducial points.

Figure 7.1 Example of input images with fiducial points

According to the difference of recording environment, the recorded data may contain different facial locations and scales. To eliminate such variation, we normalize each face. This is done by a global Procrustes analysis (GPA), followed by a histogram equalization. We will show later the improvement of this transform on the classification results.

To extract facial features we propose to calculate the coefficients of polynomial projections on a complete basis on each fiducial point. Two different modes of computation are available : coefficients can either be calculated on texture patches, or retrieved from a multi-resolution polynomial decomposition.

For the first mode - **SR_Poly**, feature vector for each facial point is extracted from a 19x19 pixels image patch centered on that point. This size was chosen to be similar to the

size calculated empirically for the approach using LBP histograms. Hence, the polynomial coefficients are obtained via projections on a 19x19 complete Hermite basis with a Chebychev function for the collocation points. Since the coefficients provide a hierarchical representation of image structures, we can reduce their number to speed-up the computations with little efficiency loss.

For the second mode- **MR_Poly**, we use a 3 level multi-resolution approach proposed in section 3.2. To have a similar representation to Gabor wavelets as [ZLSA98] we use a complete 3x3 Hermite basis with a Chebychev function for collocation points. In this way, we will have a representation with 3 scales and 9 orientations. The regions around every fiducial point vary from 81x81 pixels to 3x3 pixels. Figure 7.2 shows the first level frequency decomposition of a 3x3 polynomial approach.

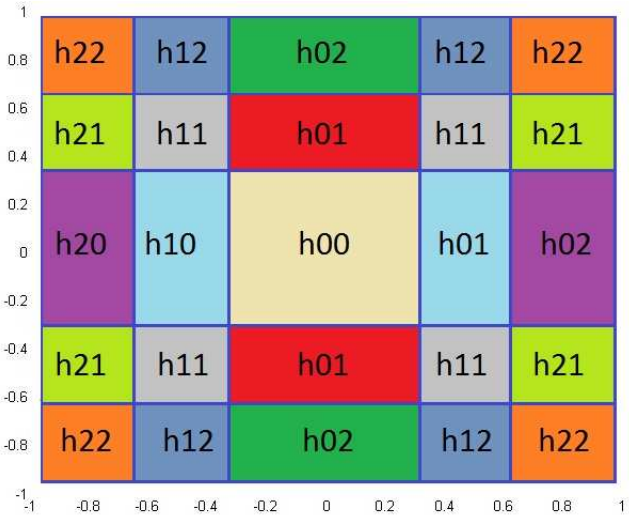


Figure 7.2 Frequency decomposition of the polynomial transform, where $h_{i,j}$ represent different subbands

Comparison with Gabor transform

Polynomial representations are similar to complete wavelet packet decompositions for a defined scale. Using a multi-resolution polynomial approach we can obtain a multi-scale/multi-orientation non-redundant representation.

Usually, in an automatic facial expression recognition system using Gabor wavelets, a bank of Gabor filters, composed of filters in distinct orientations and frequencies, is applied to the face to extract the feature vector. The filter bank is usually composed of four frequencies and six orientations. So to calculate the Gabor feature vector, each image is convolved with 24 Gabor kernels, which sizes varies with the frequencies. This representation is memory

and time consuming. For example to calculate the 6 different orientations for the biggest Gabor kernel are required $6 \times n \times n$ multiplications, n being the size of the kernel.

In 2-D the Gabor filter is defined by a two-dimensional Gaussian function modulated by a sinusoidal wave and is usually used for edge detection. The filter has a real and an imaginary component representing orthogonal directions. In this chapter we will use the real part of Gabor filters for feature extraction. The expression of the 2-D real Gabor filter is given by:

$$g(x,y,\lambda,\theta,\psi,\sigma,\gamma) = \exp(-\frac{x'^2+\gamma y'^2}{2\sigma^2})\cos(2\pi\frac{x'}{\lambda}+\psi) \tag{7.1}$$

where λ represents the wavelength of the sinusoidal factor, θ the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma/standard deviation of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

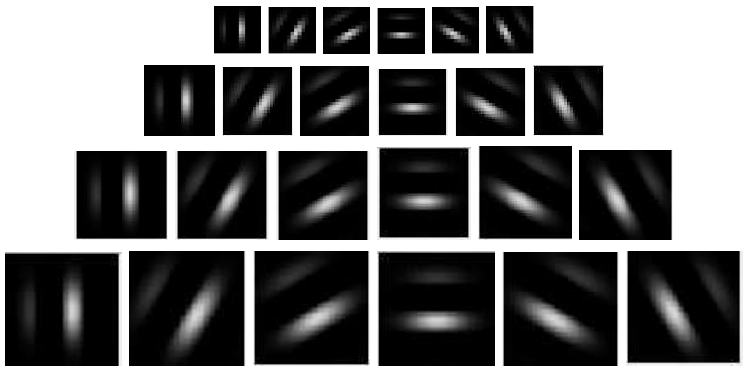


Figure 7.3 Gabor wavelets used for feature extraction

By using the polynomial projections with a 3x3 complete basis, our image patch is partitioned in 9 subblatices at each step, being considered as "orientations". Hence, the multi-scale polynomial transform will be more compact than a Gabor wavelet representation, thus allowing the disappearance of most sampling problems, such as the trade-off between orientation sampling and spatial sampling.

7.3 Experimental results

7.3.1 Experimental set-up

The Cohn-Kanade database [LCK+10] is one of the most comprehensive database in the current facial-expression-research community and consists of expression sequences of 210

adults, starting from a neutral expression and ending in the peak of the facial expression. Participants were instructed by an experimenter to perform a series of 23 facial displays, six of which were prototypical emotions including angry, disgust, fear, joy, sad and surprise. We use a subset of 115 subjects for our experiments. Only the first (neutral) and final image (the prototypical expression) of each of the selected sequences are considered for our training and testing resulting in 401 images - 45 Anger, 59 Disgust, 25 Fear, 69 Joy, 98 Neutral, 28 Sadness and 82 Surprise.

MUG database [APD10a] includes image sequences of 86 subjects performing the six basic expressions more than once. The image sequences begin and end at neutral state and follow the onset, apex, offset temporal pattern. For our experiments we used 401 images of 26 subjects that are manually annotated with 65 landmarks by removing the chin landmarks. The images are classified using the following distribution: 57 Anger, 71 Disgust, 47 Fear, 87 Joy, 25 Neutral, 48 Sadness and 66 Surprise.

To evaluate the generalization performance to novel subjects, we have adopted a 10-fold cross-validation testing scheme in our experiments. More precisely, the dataset was randomly divided into ten groups of roughly equal numbers of subjects. Nine groups were used as the training data to train classifiers, while the remaining group was used as the test data. The above process was repeated ten times for each group in turn to be omitted from the training process. As the faces in the database are frontal view, we did not consider head pose changes.

7.3.2 Normalizing images

In this part we present classification rates using original and normalized images. The showed results are obtained using single resolution polynomial projections for feature computation. The confusion matrix obtained from Cohn Kanade database using polynomial projections (SR_Poly) is presented in Table7.1.

truth pred(%)	An	Di	Fe	Ha	Ne	Sa	Su
Anger	81.4	3.3	5.3	0.0	2.9	14.8	0.0
Disgust	4.7	91.8	0.0	0.0	1.0	0.0	0.0
Fear	4.7	0.0	84.2	5.7	0.0	11.1	0.0
Happiness	0.0	1.6	10.5	92.9	1.0	0.0	0.0
Neutral	0.0	3.3	0.0	1.4	89.2	3.7	0.0
Sadness	9.3	0.0	0.0	0.0	5.9	66.7	0.0
Surprise	0.0	0.0	0.0	0.0	0.0	3.7	100

Table 7.1 Confusion matrix for the CK database (SR_Poly)

We see that happy, disgust, neutral, and surprise are detected with high accuracy while fear is presenting slightly inferior detection.

The confusion matrix obtained from MUG database using polynomial projections (SR_Poly) is presented in Table 7.2. In this case neutral presents the lowest detection, followed by fear. This is due to the low quantity of neutral images in the database. All others emotions where predicted with high accuracy.

truth pred(%)	An	Di	Fe	Ha	Ne	Sa	Su
Anger	85.7	1.4	0.0	0.0	5.0	2.0	0.0
Disgust	4.8	97.1	0.0	0.0	0.0	0.0	0.0
Fear	0.0	0.0	90.9	1.2	5.0	0.0	7.2
Happiness	1.6	1.4	2.3	96.5	0.0	0.0	1.4
Neutral	3.2	0.0	2.3	1.2	85.0	8.2	0.0
Sadness	4.8	0.0	0.0	0.0	5.0	87.8	1.4
Surprise	0.0	0.0	4.5	1.2	0.0	2.0	89.9

Table 7.2 Confusion matrix for the MUG database (SR_Poly)

As we use later for feature calculation facial points obtained using AAMs that are directly calculated in the mean face we perform the same experiments on modified images using Global Procrustes Analysis followed by a histogram normalization. The mean face is calculated independently for each database because of differences in annotation (68 versus 80 points). Some examples of the changed images are presented in Figure 7.4. It can be observed that after GPA, that includes optimal translation, rotation and scale all the faces have similar shape and are centered.

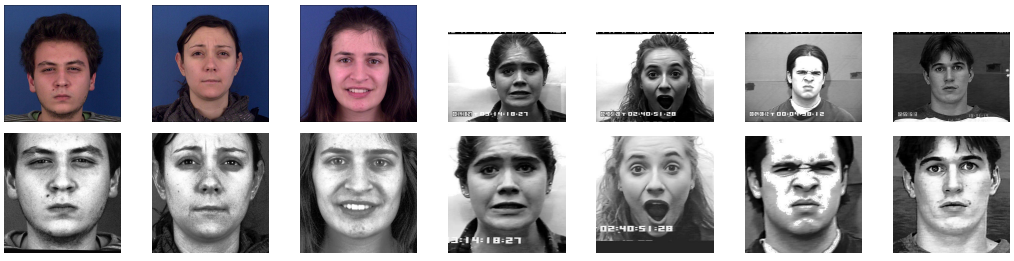


Figure 7.4 On first row - original images, on second row, modified images using Global Procrustes normalization. Left : MUG , Right : Cohn-Kanade database

The confusion matrix obtained from MUG normalized database using polynomial projections (SR_Poly) is presented in Table 7.4 and for Cohn-Kanade database in Table 7.3.

It can be observed that after normalization on Cohn Kanade database disgust, anger, happiness and surprise have an improved score. Furthermore mean classification rate increase from 89.81% to 92.05%.

truth pred(%)	An	Di	Fe	Ha	Ne	Sa	Su
Anger	87.8	0.0	4.2	0.0	4.8	13.0	0.0
Disgust	2.4	96.6	0.0	0.0	1.0	0.0	0.0
Fear	2.4	0.0	79.2	7.0	0.0	0.0	0.0
Happiness	2.4	0.0	8.3	93.0	0.0	0.0	0.0
Neutral	0.0	3.4	4.2	0.0	88.5	0.0	0.0
Sadness	4.9	0.0	0.0	0.0	5.8	87.0	0.0
Surprise	0.0	0.0	4.2	0.0	0.0	0.0	100.0

Table 7.3 Confusion matrix for the CK database (SR_Poly) using normalized images

As for MUG database it can be noticed that disgust and surprise detection is higly improved (about 10% for each expression) and the others are detected with the same rates. Classification accuracy is also enhanced from 91.52% to 94.25%.

truth pred(%)	An	Di	Fe	Ha	Ne	Sa	Su
Anger	96.6	0.0	0.0	0.0	3.8	0.0	0.0
Disgust	3.4	98.6	0.0	0.0	3.8	0.0	0.0
Fear	0.0	0.0	90.7	1.2	3.8	0.0	8.5
Happiness	0.0	1.4	2.3	97.7	0.0	0.0	1.4
Neutral	0.0	0.0	4.7	0.0	84.6	2.1	0.0
Sadness	0.0	0.0	0.0	0.0	3.8	95.8	1.4
Surprise	0.0	0.0	2.3	1.2	0.0	2.1	88.7

Table 7.4 Confusion matrix for the MUG database (SR_Poly) using normalized images

7.3.3 Comparative Study with other methods

A comparison of the proposed methods with Gabor wavelets and LBP based texture descriptors [SGM09] is shown in Figures 7.5,7.6 and Table 7.5.

Local binary pattern histograms The original LBP operator was introduced by Ojala et al., and was proved a powerful means of texture description. The operator labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the center value and considering the results as a binary number, and the 256-bin histogram of the LBP labels

computed over a region is used as a texture descriptor. [SGM09] use as descriptor histograms with 59 bins that contains only uniform patterns.(that have at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular). LBPH descriptor that we use in our tests are $LBP^{u2}_{8,2}$ descriptors computed for each key point and concatenated.

Figures 7.5 and 7.6 show the comparison results in terms of classification accuracy for all methods on annotated points with (right) and without (left) normalization, and Table7.5 the execution time for each approach. All the experiments were carried on a Dell desktop with 2.53 GHz Intel Xeon CPU. The time given for feature extraction is for one single fiducial point. It can be observed that even if *SR_poly* performs with a slightly inferior precision ($\approx 2\%$), it appears to be much better in terms of computation times w.r.t. LBP based method.

It can be observed that using normalization, all methods except Gabor wavelets on Cohn Kanade database have an increased classification rate.

Methods	Classification Rates(%)	
	Cohn Kanade	MUG
XY Positions	82.87	85.52
Gabor Wavelets	88.59	88.76
MR_Poly	89.09	90.01
LBP based method [SGM09]	95.52	91.02
SR_Poly	89.81	91.52

Figure 7.5 Comparison of proposed approaches with other methods in terms of classification accuracy using annotated points

Methods	Classification Rates(%)	
	Cohn Kanade	MUG
XY Positions	92.80	91.02
Gabor Wavelets	85.85	91.26
MR_Poly	92.55	92.76
LBPH similar to [SGM09]	96.76	94.01
SR_Poly	92.05	94.25

Figure 7.6 Comparison of proposed approaches with other methods in terms of classification accuracy using normalized annotated points

Regarding the XY positions our results differ from the one presented in [ZLSA98], where Gabor wavelets perform better than just XY positions. This is explained by the fact that our

Methods	Execution times(ms)	
	Feature extraction	Classification
XY Positions	≈ 0	0.539
Gabor Wavelets	19.637	6.810
MR_Poly	5.575	7.303
LBPH similar to [SGM09]	0.673	30.071
SR_Poly	0.411	1.420

Table 7.5 Comparison of proposed approaches with other methods in terms of execution times.

XY position are normalized by GPA, hence they are likely to give better results than Gabor wavelets.

Comparing our multiresolution polynomial approach to the one using Gabor wavelets, our method gives better performance results both in terms of accuracy as in terms of computation time. However, because the multiresolution polynomial approach implies the computation of coefficients which are unlikely to be relevant for classification, we will prefer a single-resolution method with coefficients pre-selections. As it turns out, multi-resolution decompositions are better for applications such as lossy compression or denoising than they are for classification.

It can be also observed that in comparison to the LBP based method while we obtain a slightly inferior precision ($\approx 2\%$), our method appears to be much better in terms of computation times (over twenty times faster). The length of the feature vector extracted by LBP histograms is more substantial (59 uniform patterns for each fiducial point) so in terms of classification this method is time consuming.

7.3.4 Comparative Study with other methods on calculated point with AAMs

We decided next to perform classification tests on images where key points were calculated with one of our algorithms using polynomials. We have chosen to use CDAAM method keeping 5% of polynomial coefficients for texture representation in the alignment algorithm. It should be noticed that this methos does not estimate the best keypoints (in comparison to Newton polynomial approach) but is very fast. Classification rates for the 5 approaches are presented in Figures 7.7 and 7.8.

It can be seen that using calculated points classification rates are extremely decreased for the approach using XY positions as feature and Gabor Wavelets. This can be explained

Methods	Classification Rates(%)	
	Cohn Kanade	MUG
XY Positions	57.34	65.58
Gabor Wavelets	52.58	72.82
MR_Poly	65.96	80.29
LBPH similar to [SGM09]	83.58	86.78
SR_Poly	66.95	85.04

Figure 7.7 Comparison of proposed approaches with other methods in terms of classification accuracy using calculated points

Methods	Classification Rates(%)	
	Cohn Kanade	MUG
XY Positions	58.34	61.84
Gabor Wavelets	51.65	72.58
MR_Poly	72.66	83.05
LBPH similar to [SGM09]	96.76	88.51
SR_Poly	92.05	86.29

Figure 7.8 Comparison of proposed approaches with other methods in terms of classification accuracy using normalized calculated points

by the small errors in image alignement that have a substantial significance for expression classification. Multi resolution approach using polynomials, that is comparable to Gabor wavelets present better results but have still not acceptable rates. The best outcome is obtained using LBPH method and in the normalized case followed closely by the SR_Poly approach. Considering the classification rate and the calculation times for feature extraction and classification the SR_Poly approach can be considered as the one giving the best compromise.

7.4 Discussion and conclusion

In this chapter, we have proposed a new method of using coefficients obtained by polynomial projections for recognition of expressions from still face images.

We have shown that polynomial multi-resolution decomposition allows hierarchical organization of image information within the frequency domain. As a result, polynomial coefficients can be used as an efficient alternative to global or redundant texture representations such as Gabor Wavelets, without losing accuracy. Because polynomials in the complete basis are orthogonal, it is possible to compute the coefficients directly by a simple inner product of

polynomials with the image. In multi-scale complete basis decompositions, while perfect reconstruction of the original signal can be obtained using a full set of coefficients, scalable approximation is also possible, by restricting reconstruction to a reduced set of coefficients, resulting in a fully scalable process.

Experimental results confirm that our approach performs well with face expression recognition, giving high accuracy results and being computationally efficient when facial key points are manually labelled or calculated via an alignement algorithm.

Chapter 8

Conclusion

In this thesis we explored novel techniques to use polynomial projection basis for facial analysis. Our work was motivated by the objective of proposing new facial texture representations to improve Active Appearance Models rending them robust to various factors such as physiognomies, illumination effects or poses.

In section 4.2, we show two approaches of using polynomial coefficients as a texture representation in active appearance models. The methods show that it is possible to achieve good alignment results without using global or redundant texture representations such as PCA on pixel intensities or Gabor wavelets. By construction, the polynomials in the complete basis are orthogonal and hierarchically organized, allowing fast computation and precise localization of facial landmarks. The resulting approaches are tested on various datasets and experimental results show that our approaches perform very well with face alignment algorithms and depending on the chosen method we obtain robustness to pose changes or facial expression changes.

To reduce the model complexity the proposed framework is then extended to image compression (section 4.3). The contributions at this level are twofold. First, we show how polynomial compression can be integrated into the AAM framework. This type of texture representation is multi-scale and can be used for analysis/synthesis at any given scale (provided by the polynomial degree). The selection of the strongest energy coefficients allows very stable alignment results even for high compression ratios and a good quality of the synthesized images. This shows the capacity of polynomial coefficients to represent an image in a sparse and compact manner. Second, in order to benefit of the great speed and accurate results of the regression based facial alignment approaches we include the approximation coefficients in a regression iterative framework. By conducting an experiment using seven compression ratios we demonstrated that the polynomial representation offers

advantages over the traditional Haar and CDF 9/7 wavelet in terms of synthesis quality and accuracy.

In Chapter 5, we take advantage of the polynomial property stating that multiresolution polynomial decomposition approach is equivalent to a filter bank , allowing to use the projection coefficients in a gradient descent algorithm. By reformulating the inverse compositional algorithm to entertain fitting across multiple filter responses and using first and second order polynomials, we adapt the Gauss-Newton and Newton algorithms. The method is tested on different datasets and the gain provided by the use of polynomials is substantial for all of them and overcomes the small extra time spent for polynomial projections. To our knowledge the proposed approach is the first unified solution dealing with gradient descent and texture representation into a single consistent model.

Considering that polynomial bases have been used to detect and characterize singularities in a vector field in Chapter 6 we propose to use them for an accurate keypoint localization in the image domain. The proposed algorithm consists in the calculation of a vector field followed by the research on interesting points, both presented in the context of multi-scale and multi-resolution scheme. The approach is extensively tested on the Oxford dataset and our approach is showed to be robust to scale, illumination and blur changes. Furthermore we use the selected regions by our detector in an AAM using a sparse texture model that demonstrates the quality of the selected keypoints.

In Chapter 7, we explore the use of the polynomial representation for extracting information about facial expressions. The main idea is to use as descriptor the polynomial projections around facial keypoints that are hand-labbeled or directly calculated by one of our previous AAM algorithms. Experimental results confirm that our approach performs well with face expression recognition, being computationally efficient and giving high accuracy results.

8.1 Perspectives

The perspectives regarding the work presented in this thesis can be divided in short term and long term objectives. In the short term, many improvements can be considered:

- Regarding the Chapter 4.2 a hybrid approach combining the two proposed methods can be developed to improve the fitting accuracy when the database presents variations both in pose and identity.

- Estimating the necessary degree of polynomials when using the polynomial texture representation. The size of the polynomial basis should be computed to limit the size of

the detected faces to avoid computational cost while giving the same accuracy results as the one presented above.

- Identifying the appropriate appearance descriptors for the Chapter 6, where we detect interest keypoints. Since the majority of image detectors comes with an adapted descriptor, we believe that providing a specific polynomial descriptor can greatly improve the matching experiments.

- Further testing of the proposed approaches for into-the-wild databases. A crucial point is to identify the best approach to be used with uncontrolled environments.

In the long term, there are more challenging objectives that naturally derive from this work:

- Exploring the possibility of extending the AAM model to color or multispectral images using color based polynomials namely Hypercomplex 2D polynomial basis. Compared to complete bases the hypercomplex polynomial transform has an extra parameter μ, a unit imaginary quaternion which corresponds to a privileged direction of analysis. Typically μ is the unit quaternion giving the gray axis, but for multi channel images this direction is calculated via Principal Component Analysis. The color pixels correspond to a 3D point clouds whose structure reflects the colors. The shape of this point cloud can be approximately captured with a 3D ellipsoid whose axis are the eigenvectors of the covariance matrix of the points cloud (i.e. of the pixels). The ellipsoid direction are obtained with a eigen decomposition of the symmetric covariance matrix. In the Hypercomplex polynomial decomposition, the local direction is the eigenvector associated with the higher eigenvalue.

- Finally a clear direction still left is exploring the use of 3D polynomial bases for 3D deformable models. Polynomial bases can be used for optical flow estimation therefore a model describing the texture and motion must be developed within a 3D AAM model.

Bibliography

[ABD12] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE features. In *Eur. Conf. on Computer Vision (ECCV)*, 2012.

[ABV09] Brian Amberg, Andrew Blake, and Thomas Vetter. On compositional image alignment, with an application to active appearance models. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1714–1721. IEEE, 2009.

[AC12] Bertrand Augereau and Philippe Carré. Hypercomplex polynomial wavelet packet application for color image. In *AGACSE 2012 Proceedings - Applied Geometric Algebras in Computer Science and Engineering (AGACSE 2012), La Rochelle : France*, 2012.

[AiMZ] Joan Alabort-i Medina and Stefanos Zafeiriou. Bayesian active appearance models.

[AKB08] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision–ECCV 2008*, pages 102–115. Springer, 2008.

[ANB13] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *British Machine Vision Conf. (BMVC)*, 2013.

[APD10a] N. Aifanti, C. Papachristou, and A. Delopoulos. The mug facial expression database. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4. IEEE, 2010.

[APD10b] N. Aifanti, C. Papachristou, and A. Delopoulos. The mug facial expression database. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4. IEEE, 2010.

[AV11] Brian Amberg and Thomas Vetter. Optimal landmark detection using shape models and branch and bound. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 455–462. IEEE, 2011.

[AZCP13] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3444–3451. IEEE, 2013.

[BAPD13] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1513–1520. IEEE, 2013.

[Bet12] V. Bettadapura. Face expression recognition and analysis: the state of the art. *arXiv preprint arXiv:1203.6722*, 2012.

[BJKK11] Peter N Belhumeur, David W Jacobs, D Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 545–552. IEEE, 2011.

[Bla74] AS Blaivas. Visual analysis in unspecialized receptive fields as an orthogonal series expansion. *Neurophysiology*, 6(2):168–173, 1974.

[BLF+05] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE, 2005.

[BM02] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-02-16*, 2002.

[BM04] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[BML+01] Hans G Bosch, Steven C Mitchell, Boudewijn PF Lelieveldt, Francisca Nijland, Otto Kamp, Milan Sonka, and Johan HC Reiber. Active appearance motion models for endocardial contour detection in time sequences of echocardiograms. In *Medical Imaging 2001*, pages 257–268. International Society for Optics and Photonics, 2001.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[BV03] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003.

[CC06] David Cristinacce and Timothy F Cootes. Feature detection and tracking with constrained local models. In *BMVC*, volume 2, page 6, 2006.

[CC07] David Cristinacce and Timothy F Cootes. Boosted regression active shape models. In *BMVC*, pages 1–10, 2007.

[CC08] David Cristinacce and Tim Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

[CCTC09] Angela Caunce, David Cristinacce, Chris Taylor, and Tim Cootes. Locating facial features and pose estimation using a 3d shape model. In *Advances in Visual Computing*, pages 750–761. Springer, 2009.

[CET99] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Comparing active shape models with active appearance models. In *BMVC*, volume 99, pages 173–182, 1999.

[CET01a] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[CET01b] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[CILS12] Tim F Cootes, Mircea C Ionita, Claudia Lindner, and Patrick Sauer. Robust and accurate shape model fitting using random forest regression voting. In *Computer Vision–ECCV 2012*, pages 278–291. Springer, 2012.

[CP12] DG Ciric and VD Pavlovic. Linear phase two-dimensional fir digital filter functions generated by applying christoffel-darboux formula for orthonormal polynomials. *Electronics and Electrical Engineering*, 120(4):39–42, 2012.

[CT01] T.F. Cootes and C.J. Taylor. On representing edge structure for model matching. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1114. IEEE, 2001.

[CT+04] Timothy F Cootes, Christopher J Taylor, et al. Statistical models of appearance for computer vision, 2004.

[CT06] Timothy F Cootes and Christopher J Taylor. An algorithm for tuning an active appearance model to new data. In *BMVC*, pages 919–928. Citeseer, 2006.

[CTCG95] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[CW07] B.W. Char and S.M. Watt. Representing and characterizing handwritten mathematical symbols through succinct functional approximation. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 1198–1202. IEEE, 2007.

[CWLZ13] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Trans. Graph.*, 32(4):41, 2013.

[CWWS14] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[CWWT02] Timothy F Cootes, Gavin V Wheeler, Kevin N Walker, and Christopher J Taylor. View-based active appearance models. *Image and vision computing*, 20(9):657–664, 2002.

[Dau85] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169, 1985.

[DAVM04] F. Davoine, B. Abboud, and D. VAN MO. Analyse de visages et d'expressions faciales par modèle actif d'apparence. *TS. Traitement du signal*, 21(3):179–193, 2004.

[DB06] Michael Donoser and Horst Bischof. Efficient maximally stable extremal region (mser) tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 553–560. IEEE, 2006.

[DGFVG12] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Real-time facial feature detection using conditional regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2578–2585. IEEE, 2012.

[DWP10] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1078–1085. IEEE, 2010.

[EAS11] F. Ernawan, N.A. Abu, and N. Suryana. Spectrum analysis of speech recognition via discrete tchebichef transform. In *2011 International Conference on Graphic and Image Processing*, pages 82856L–82856L. International Society for Optics and Photonics, 2011.

[EFM09] Micah Eckhardt, Ian Fasel, and Javier Movellan. Towards practical facial feature detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(03):379–400, 2009.

[ESZ06] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy–automatic naming of characters in tv video. 2006.

[ETC98] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, FG '98, pages 300–, Washington, DC, USA, 1998. IEEE Computer Society.

[EUL86] M. Eden, M. Unser, and R. Leonardi. Polynomial representation of pictures. *Signal Processing*, 10(4):385–393, 1986.

[FGMR10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[FH05] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[GF14] Golnaz Ghiasi and Charless C Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1899–1906. IEEE, 2014.

[GMB05] Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

[GMB06] Ralph Gross, Iain Matthews, and Simon Baker. Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604, 2006.

[HDF12] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative Evaluation of Binary Features. In *European Conference on Computer Vision (ECCV)*, 2012.

[HFT+03] C. Hu, R. Feris, M. Turk, et al. Active wavelet networks for face alignment. In *British machine vision conference*, 2003.

[HM09] Onur C Hamsici and Aleix M Martinez. Active appearance models with rotation invariant kernels. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1003–1009. IEEE, 2009.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[K+12] R. Krishnamoorthi et al. A multiresolution approach for rotation invariant texture image retrieval with orthogonal polynomials model. *Journal of Visual Communication and Image Representation*, 23(1):18–30, 2012.

[KJ14] Vahid Kazemi and Sullivan Josephine. One millisecond face alignment with an ensemble of regression trees. 2014.

[KK09] R. Krishnamoorthi and N. Kannan. A new integer image coding technique based on orthogonal polynomials. *Image and vision computing*, 27(8):999–1006, 2009.

[KnC06] P. Kittipanya-ngam and TF Cootes. The effect of texture representations on aam performance. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 328–331. IEEE, 2006.

[KSS11] Ira Kemelmacher-Shlizerman and Steven M Seitz. Face reconstruction in the wild. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1746–1753. IEEE, 2011.

[KSYY10] Tatsuo Kozakaya, Tomoyuki Shibata, Mayumi Yuasa, and Osamu Yamaguchi. Facial feature localization using weighted vector concentration approach. *Image and Vision Computing*, 28(5):772–780, 2010.

[KTAK10] Olivier Kihl, Benoit Tremblais, Bertrand Augereau, and Majdi Khoudeir. Human activities discrimination with motion approximation in polynomial bases. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2469–2472. IEEE, 2010.

[LCK+10] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[LH15] Gil Levi and Tal Hassner. LATCH: learned arrangements of three patch codes. *CoRR*, abs/1501.03719, 2015.

[Liu07] Xiaoming Liu. Generic face alignment using boosted appearance model. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[Liu10] Xiaoming Liu. Video-based face model fitting using adaptive active appearance model. *Image and Vision Computing*, 28(7):1162–1172, 2010.

[LMH+06] Simon Lucey, Iain Matthews, Changbo Hu, Zara Ambadar, Fernando De la Torre, and Jeffrey Cohn. Aam derived face representations for robust facial action recognition. 2006.

[Low99] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[LSD+07] Rasmus Larsen, Mikkel B Stegmann, Sune Darkner, Søren Forchhammer, Timothy F Cootes, and Bjarne Kjær Ersbøll. Texture enhanced appearance models. *Computer Vision and Image Understanding*, 106(1):20–30, 2007.

[Mar06] J.B. Martens. The hermite transform: a survey. *EURASIP Journal on applied signal processing*, 2006:97–97, 2006.

[MB04] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[MBN13] Stephen Milborrow, Tom Bishop, and Fred Nicolls. Multiview active shape models with sift descriptors for the 300-w face landmark challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 378–385, 2013.

[MN08] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *Computer Vision–ECCV 2008*, pages 504–513. Springer, 2008.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[MVBP13] Brais Martinez, Michel François Valstar, Xavier Binefa, and Maja Pantic. Local evidence aggregation for regression-based facial point detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5):1149–1163, 2013.

[ND09] Srinivas Nagamalla and Bibhas Chandra Dhara. A novel face recognition method using facial landmarks. In *Advances in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference on*, pages 445–448. IEEE, 2009.

[NLSS04] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM face database - an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, may 2004.

[NSL11] Rajitha Navarathna, Sridha Sridharan, and Simon Lucey. Fourier active appearance models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1919–1926. IEEE, 2011.

[PM08] George Papandreou and Petros Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[PPB08] Daniel Pizarro, Julien Peyras, and Adrien Bartoli. Light-invariant fitting of active appearance models. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6. IEEE, 2008.

[RBDlT+11] Gemma Roig, Xavier Boix, Fernando De la Torre, Joan Serrat, and Carles Vilella. Hierarchical crf with product label spaces for parts-based models. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 657–664. IEEE, 2011.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[RGP+99] Sami Romdhani, Shaogang Gong, Ahaogang Psarrou, et al. A multi-view nonlinear active shape model using kernel pca. In *BMVC*, volume 10, pages 483–492, 1999.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[RSBP11] Vincent Rapp, Thibaud Senechal, Kevin Bailly, and Lionel Prevost. Multiple kernel learning svm and statistical validation for facial landmark detection. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 265–271. IEEE, 2011.

[Sad96] I. Sadeh. Polynomial approximation of images. *Computers & Mathematics with Applications*, 32(5):99–115, 1996.

[SB97] Stephen M Smith and J Michael Brady. Susan - a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[SBB03] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression database. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1615–1618, 2003.

[SFC04] M. B. Stegmann, S. Forchhammer, and T. F. Cootes. Wavelet enhanced appearance modelling. In *SPIE International Symposium on Medical Imaging*, volume 5370, pages 1823–1832, San Diego, CA, 2004. SPIE.

[SG07] Jason Saragih and Roland Goecke. A nonlinear discriminative approach to aam fitting. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[SGM09] C. Shan, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.

[SK09] Jaewon Sung and Daijin Kim. Adaptive active appearance model with incremental learning. *Pattern Recognition Letters*, 30(4):359–367, 2009.

[SL03] M.B. Stegmann and R. Larsen. Multi-band modelling of appearance. *Image and Vision Computing*, 21(1):61–67, 2003.

[SLBW13] Xiaohui Shen, Zhe Lin, Jonathan Brandt, and Ying Wu. Detecting and aligning faces by image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3460–3467. IEEE, 2013.

[SLC11] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[SLGBG09] Renaud Séguier, Sylvain Le Gallou, Gaspard Breton, and Christophe Garcia. Adapted active appearance models. *EURASIP Journal on Image and Video Processing*, 2009(1):945717, 2009.

[SS09] Keshav Seshadri and Marios Savvides. Robust modified active shape model for automatic facial landmark annotation of frontal faces. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, pages 1–8. IEEE, 2009.

[SS12] Keshav Seshadri and Marios Savvides. An analysis of the sensitivity of active shape models to initialization when applied to automatic facial landmarking. *Information Forensics and Security, IEEE Transactions on*, 7(4):1255–1269, 2012.

[STLG09] Y. Su, D. Tao, X. Li, and X. Gao. Texture representation in aam using gabor wavelet and local binary patterns. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 3274–3279. IEEE, 2009.

[TAiMZP13] Georgios Tzimiropoulos, Joan Alabort-i Medina, Stefanos Zafeiriou, and Maja Pantic. Generic active appearance models revisited. In *Computer Vision–ACCV 2012*, pages 650–663. Springer, 2013.

[TLWT12] Yan Tong, Xiaoming Liu, Frederick W Wheeler, and Peter H Tu. Semi-supervised facial landmark annotation. *Computer Vision and Image Understanding*, 116(8):922–935, 2012.

[TP13] Georgios Tzimiropoulos and Maja Pantic. Optimization problems for fast aam fitting in-the-wild. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 593–600. IEEE, 2013.

[UFH12] Michal Uřičář, Vojtěch Franc, and Václav Hlaváč. Detector of facial landmarks learned by the structured output svm. *VISAPP*, pages 547–556, 2012.

[VJ01] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[VJ04] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[VMBP10] Michel Valstar, Brais Martinez, Xavier Binefa, and Maja Pantic. Facial point detection using boosted regression and graph models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2729–2736. IEEE, 2010.

[WFKVDM97] Laurenz Wiskott, J-M Fellous, N Kuiger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):775–779, 1997.

[WFY12] T. Wu, S. Fu, and G. Yang. Survey of the facial expression recognition research. In *Advances in Brain Inspired Cognitive Systems*, pages 392–402. Springer, 2012.

[WGTL14] Nannan Wang, Xinbo Gao, Dacheng Tao, and Xuelong Li. Facial feature point detection: A comprehensive survey. *arXiv preprint arXiv:1410.1037*, 2014.

[WLD08] Hao Wu, Xiaoming Liu, and Gianfranco Doretto. Face alignment via boosted ranking model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[WT99] CBH Wolstenholme and Christopher J Taylor. Wavelet compression of active appearance models. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI'99*, pages 544–554. Springer, 1999.

[XDlT13] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 532–539. IEEE, 2013.

[YHL+03] Shuicheng Yan, Xinwen Hou, Stan Z Li, Hongjiang Zhang, and Qiansheng Cheng. Face alignment using view-based direct appearance models. *International journal of imaging systems and technology*, 13(1):106–112, 2003.

[YHZ+13] Xiang Yu, Junzhou Huang, Shaoting Zhang, Wang Yan, and Dimitris N Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1944–1951. IEEE, 2013.

[YLYL13] Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Learn to combine multiple hypotheses for accurate face alignment. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 392–396. IEEE, 2013.

[YR11] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[ZA06] Li Zhang and Haizhou Ai. Multi-view active shape model with robust parameter estimation. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 469–468. IEEE, 2006.

[ZBL13] Feng Zhou, Jonathan Brandt, and Zhe Lin. Exemplar-based graph matching for robust facial landmark localization. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1025–1032. IEEE, 2013.

[ZLSA98] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 454–459. IEEE, 1998.

[ZR12] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.

[ZSCC13] Xiaowei Zhao, Shiguang Shan, Xiujuan Chai, and Xilin Chen. Locality-constrained active appearance model. In *Computer Vision–ACCV 2012*, pages 636–647. Springer, 2013.