Optimisation hybride mono et multi-objectifs de modèles actifs d'apparence 2,5D pour l'analyse de visage

Abdul Sattar

N° d'ordre : 4029 **ANNÉE 2010**

UNIVERSITÉ DE RENNES 1

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du Signal et Télécommunications

Ecole doctorale : Matisse

présentée par

Abdul SATTAR

Préparée à l'unité de recherche : SCEE-SUPELEC/IETR UMR 6164
Nom développé de l'unité : Institut d'Electronique et de Télécommunications de Rennes
Composante universitaire : S.P.M.

Face Analysis by using Hybrid Single Objective and Hybrid Multiple Objective Optimizations in 2.5D Active Appearance Model

Thèse soutenue à Supélec-Rennes le 29 Avril 2010
devant le jury composé de :

Mireille GARREAU
Examinateur
Professeur, LTSI, Université de Rennes 1, Rennes

Gaspard BRETON
Examinateur
Ingénieur R&D, France Telecom, Rennes

Saida BOUAKAZ
Rapporteur
Professeur, LIRIS, Lyon

Clarisse DHAENENS
Rapporteur
Professeur, Polytech'Lille, INRIA, Université de Lille I

Luce MORIN
Membre Invité
Professeur, IETR, Rennes

Jacques PALICOT
Directeur
Professeur, Supélec/IETR, Rennes

Renaud SEGUIER
Co-Directeur
Professeur adjoint, Supélec/IETR, Rennes

Acknowledgments

I would like the readers to keep all these people in mind while reading this dissertation, without them this success would not be possible.

I would like to begin by thanking Renaud Seguier for his guidance, understanding, patience, always believing in my capabilities and most importantly, his friendship during my PhD at Supélec. He acted as a boss, advisor, friend and brother to me from time to time. He has always encouraged me to not only grow as an instructor but also as a developer and an independent thinker. It is my honour to work with him.

I would also like to pay my sincere gratitude to Jacques Palicot for giving me an opportunity to work in SCEE team. I can say SCEE is a great place to do research work because of its friendly work environment, open policies, and overall culture. I thank all my colleagues, administration and 5050 team of Supélec for creating such a pleasant work environment and for being there for me. In particular many thanks to Sylvain Le Gallou and Nicolas Stoiber with whom I worked closely and had many fruitful technical discussions. Many thanks to Olivier Aubault and Gaspard Breton for providing me the opportunity to implement my thesis work in real time.

Most importantly I would like to thank my late parents, Abdullah and Saira, for their faith in me and allowing me to be as ambitious as I wanted. It was under their watchful eye that I gained so much drive and an ability to tackle challenges head on.

A very special thanks goes to my siblings, wife Naheed, son Muhammad Uzair and daughter Maryam. I cannot forget their support, motivation, patience and sacrifices they made especially during periods of intense workload. Also I thank to all my friends in France and Pakistan for motivating me and regularly wishing me good luck.

I am very grateful to all those people who made me feel at home, cared for and allowed me to worry only about my studies during my stay in France. Honestly this list is huge, but I would like to mention secretaries and technical staff of Supélec, Mr. Touffet of HLM residence, administrative staff of Cesson Sévigné and teachers of Centre d'Etude de Langue of Colmar.

Finally I would like to express my indebtedness to all the jury members for spending their precious time to read and accept my work.

This dissertation is dedicated to my wife, kids and late parents.

Contents

Thesis Summary in French

1 Introduction

L'équipe SCEE de Supélec travaille dans le domaine de la radio logicielle et intelligente, encore appelée Radio Cognitive (CR - Cognitive Radio). Dans cette thèse, nous avons présenté une solution pour l'analyse de visage temps réel dans un équipement de radio cognitive. Un équipement CR intègre en plus des composants radio classiques, un ensemble de capteurs intelligents. A partir des informations fournies par ces capteurs, le CRM (Cognitive Radio Manager) va définir la configuration optimale afin de fournir le meilleur service. La fonction du capteur vidéo est, entre autre, d'authentifier l'utilisateur de l'équipement et de détecter ses caractéristiques faciales en plus de l'orientation de son visage. Ces informations guident le choix du codec vidéo le mieux adapté, l'objectif final étant de fournir la meilleur qualité de transmission compte tenu du contexte global dans lequel évolue l'équipement de Radio Cognitive. Dans ce cadre particulier, nous proposons des solutions d'analyse de visage, à savoir *l'estimation de la pose et des caractéristiques faciale d'un visage inconnu orienté.*

L'analyse faciale est à considérer au sens large et inclus l'alignement de visage, l'estimation de sa pose et enfin l'extraction de caractéristiques et expressions de visages inconnus. Pour l'extraction de la pose du visage dans un espace à six degrés de liberté, le paramètre le plus problématique est la rotation en dehors du plan (quand on fait "non" de la tête, l'angle qui correspond au "yaw" ou au "lacet"), comparé aux autres paramètres de rotation et translation. De plus, si le visage est inconnu du système, la détection s'avère plus difficile à réaliser. Notre travail de thèse a consisté à proposer une solution pour la détection de cet angle de rotation en dehors du plan pour des visages inconnus.

Les visages humains sont des objets par nature non-rigides. Le problème de cette flexibilité est pris en compte dans les Modèles Actifs d'Apparence (AAM) [1] qui sont remarquablement efficaces lorsqu'il s'agit d'extraire des caractéristiques faciales et plus généralement lorsqu'il faut aligner un visage (opération qui consiste à localiser plusieurs dizaines de points autour des yeux, du nez, de la bouche et des sourcils). Notre système d'analyse de visage est basé sur de nouveau modèles actifs d'apparence en 2.5D qui s'appuient sur une optimisation hybride mono et multi-objectifs.

La solution hybride est nécessaire étant donné la forme non-convexe de l'erreur dans l'espace de recherche multidimensionnel générée par les AAM. Tant que le visage analysé reste de face tout en étant correctement localisé, l'erreur entre le modèle

d'apparence et le visage réel reste convexe. Pour peu qu'il se déplace latéralement, des minimum locaux apparaissent sur cette surface. Cette perte de convexité rend nécessaire l'utilisation d'algorithmes d'optimisation ayant d'égales capacités d'exploration et d'exploitation. Nous entendons par exploration la capacité à trouver une solution globale n'importe où dans l'espace de recherche, et par exploitation la capacité à utiliser des informations issues de solutions préalables afin d'améliorer les solutions futures proposées par l'algorithme d'optimisation. Les algorithmes génétiques (GA) sont souvent utilisés comme des algorithmes de recherche globale compte tenu de leur qualité d'exploration, tandis que la descente de gradient (GD) qui trouve facilement des optimums locaux peut aider les GA à améliorer leur capacité d'exploitation. En d'autres termes, les capacités d'exploration des GA et d'exploitation des GD peuvent mener à un algorithme d'optimisation hybride efficace.

Ce problème de la non-convexité de la surface de l'erreur dans l'espace des paramètres de l'AAM est non seulement abordé par la proposition d'un algorithme hybride, mais aussi par la prise en compte de plusieurs webcams. Historiquement cette solution était inenvisageable compte tenu du prix des caméras et de la faible puissance des processeurs devant traiter le flot des données issus de ces caméras. Actuellement, étant donné le prix dérisoire des webcams et l'augmentation de la capacité des processeurs, un tel système est tout à fait viable.

Dans un système mono-vue, l'alignement d'un visage ne peut être envisagé lorsqu'une partie du visage masque littéralement une autre partie comme cela arrive lors d'une rotation de type "yaw", par exemple dans le cas d'une vue de profil. Pour contourner ce problème, nous utilisons et réalisons la fusion des informations venant de plusieurs caméras. Des informations mêmes partielles issues de plusieurs caméras aident l'alignement : dans un système multi-vues, plus on a de sources d'information, plus le système est robuste : la probabilité de trouver des solutions divergentes c'est à dire fortement mal alignées, est alors minime.

La capture de plusieurs images d'un même visage mène à la production de plusieurs surfaces caractérisant l'erreur d'alignement. La recherche d'une solution optimale concernant un processus unique lui-même caractérisé par plusieures fonctions d'erreur conduit naturellement à l'optimisation multi-objectifs ***MOO - Multi Objectives Otimization)***. Beaucoup de techniques de type MOO existent mais nous avons choisi l'algorithme bien connu NSGA-II basé sur l'approche de Pareto pour ses qualités reconnues en exploration et son efficacité éprouvée dans le domaine de la modélisation des contours de la bouche [2].

Nous proposons deux systèmes d'alignement de visages. 1) Le premier exploite un AAM 2.5D et une seule caméra. La phase d'optimisation de cet AAM est hybride : elle mixe un algorithme génétique et une descente de gradient. Notre contribution tient dans l'opérateur de descente de gradient qui travaille de concert avec l'opérateur classique de mutation : de cette manière sa présence ne pénalise pas la vitesse d'exécution du système. 2) Le second met en œuvre un AAM 2.5D mais exploite plusieurs caméras. La recherche de la meilleure solution découle également d'une approche hybride qui mixe une optimisation multi-objectifs : le NSGA-II, avec une descente de gradient. Notre contribution tient dans la proposition d'une méthode efficace pour extraire des

informations concernant la pertinence de chacune des vues, ces informations sont ensuite exploitées par la descente de gradient.

2 Modèle Actif d'Apparence 2.5D

Cette section présente notre première contribution qui consiste en la génération d'un modèle actif d'apparence 2.5D. Nous l'utiliserons pour estimer la pose et extraire les caractéristiques faciales d'un visage produisant des mouvements de rotation latérale de forte amplitude. L'AAM 2.5D est construit à partir i) de marqueurs 2D positionnés sur une vue de face et de profil du visage et permettant de générer un modèle 3D et ii) d'une texture 2D extraite à partir de la vue de face et projetée sur le modèle 3D du visage. Dans cette phase de modélisation (voir figure 1), 68 marqueurs sont spécifiés manuellement sur l'ensemble des images de la base d'exemples de visages.

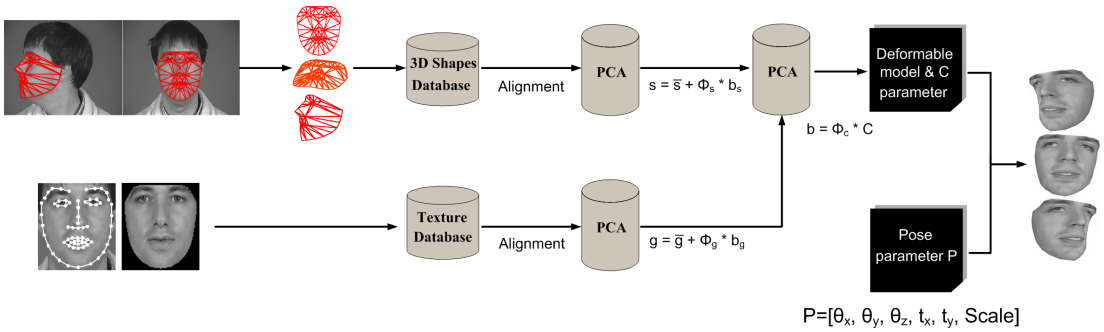


Figure 1: Modélisation AAM 2.5D d'un visage

Les marqueurs de tous les exemples de visages sont normalisés et alignés dans les trois dimensions à partir d'une analyse de Procruste ([3], [4]). La moyenne de ces marqueurs normalisés constitue la forme moyenne des visages existants dans la base. On associe donc à chaque exemple de visage une forme spécifique 3D produite à partir de ces marqueurs. Une Analyse en Composantes Principale (PCA) est mise en œuvre sur l'ensemble des formes pour produire un vecteur de paramètres de forme qui pourra représenter 95% des variations de formes de la base d'exemples.

$$s_i = \bar{s} + \phi_s * b_s \tag{1}$$

avec s_i la forme synthétisée, $\bar{s}$ la forme moyenne, ϕ_s les vecteurs propres produits par la PCA et b_s les paramètres de formes.

Les vues de faces de tous les visages contenus dans la base sont projetés ("*warpés*") sur la forme moyenne 3D obtenue dans la phase précédente. Les textures *warpées* sont projetées sur un plan 2D et génèrent ainsi des vues frontales 2D de toutes les textures des visages de la base d'apprentissage dans une même forme moyenne de visage. C'est pour cette raison que nous nommons notre modèle un AAM 2.5D : il résulte de marqueurs définis dans l'espace 3D qui génèrent une forme 3D sur laquelle vient se projeter une texture 2D. La moyenne des textures est évaluée à partir de l'ensemble des textures

warpées sur la forme moyenne. Sur cet ensemble est appliqué une seconde PCA qui va produire un vecteur de paramètres caractérisant 95% des variations des textures de la base.

$$g_i = \bar{g} + \phi_g * b_g \tag{2}$$

où g_i est la texture synthétisée, $\bar{g}$ est la texture moyenne, ϕ_g sont les vecteurs propres produits par la PCA et b_g les paramètres de texture.

Finalement les vecteurs b_s et b_g de chaque exemple de visage sont concaténés pour former un vecteur b. Sur l'ensemble de ces vecteurs est réalisée une dernière PCA :

$$b = [b_s b_g]^T, b = \phi_C * C \tag{3}$$

où ϕ_C représente les vecteurs propres caractérisant 95% de la variation totale des données et C est le vecteur des paramètres d'apparence qui va définir à la fois la forme et la texture de chaque visage de la base d'exemples.

Le modèle 2.5D peut être modifié en translation, en rotation et en zoom par l'intermédiaire du vecteur de pose P.

$$P = [\theta_{pitch}, \theta_{yaw}, \theta_{roll}, t_x, t_y, Scale]^T \tag{4}$$

où θ_{pitch} correspond à la rotation du visage autour de l'axe x (lorsque l'on fait oui de la tête), θ_{yaw} est associé à une rotation autour de l'axe y (pour faire un profil ou un semi-profil) et θ_{roll} correspond à la rotation dans le plan. t_x, t_y représentent la position du centre du visage et $Scale$ et un facteur de zoom. La figure 2 illustre le modèle lorsqu'on le fait tourner d'un angle θ_{yaw}.

Figure 2: Le modèle AAM 2.5D pour différentes valeurs de θ_{yaw}

Dans la phase de segmentation, le modèle d'un visage déformé et modifié en translation, zoom et en rotation par l'intermédiaire des vecteurs C et P est positionné sur l'image analysée I. La texture segmentée de cette image (délimitée par les points du modèle) est projetée (*warpée*) dans la forme moyenne des visages, puis normalisée en luminance pour éviter les problèmes de variations lumineuse. L'objectif est de minimiser l'erreur pixel e suivante.

$$e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[I_i(C,P) - M_i(C)]^2} \tag{5}$$

où $I(C,P)$ est l'image segmentée et $M(C)$ est la texture du modèle générée par le vecteur C. N est le nombre de pixels de la texture. Pour choisir les bons paramètres C et

P nous avons besoin d'un algorithme d'optimisation visant à minimiser l'erreur e. Dans notre proposition associée à un système mono-vue, ces deux vecteurs sont optimisés conjointement par un algorithme génétique hybride tandis que pour le système multi caméras, c'est un NSGA-II hybride qui sera implémenté.

3 Système mono-vue

Cette section présente notre contribution consistant en un algorithme d'optimisation hybride pour les AAM qui intègre une descente de gradient dans un algorithme génétique dans le but de réaliser un alignement de visage robuste, efficace et fonctionnant en temps réel.

3.1 Optimisation

Un algorithme génétique peut être utilisé pour optimiser la valeur des vecteurs d'apparence C et de pose P. L'objectif est de trouver les meilleures valeurs possibles de ces vecteurs de paramètres pour minimiser l'erreur entre le modèle et l'image analysée. Dans ce cadre, chaque paramètre est considéré comme un gène. Tous les gènes de C et P sont concaténés et constituent un chromosome. Une population d'un certain nombre de ces chromosomes est créée de façon aléatoire. La somme e (Eq.5) des erreurs pixel (la fitness) entre la texture du modèle produit par le chromosome et la texture de l'image analysée est évaluée. Les chromosomes pris en compte dans la reproduction sont sélectionnés par une procédure de type Tournoi. Un crossover sur deux points et une mutation gaussienne sont appliqués pour produire la nouvelle population de chromosomes.

3.2 Algorithme hybride

La fusion d'un AG et d'une descente de gradient n'est pas évidente compte tenu de la nature très différente de ces deux algorithmes. Une fusion rudimentaire augmenterait le nombre de calculs d'erreurs. Nous proposons un *opérateur gradient* qui travaille de concert avec l'opérateur de mutation. Ce nouvel opérateur utilise l'erreur évaluée durant la mutation et ne nécessite donc pas d'évaluation supplémentaire de cette erreur. Le paragraphe suivant détaille notre approche.

Operateur Gradient Durant la mutation d'un chromosome dans le GA nous avons changé la valeur d'un seul gène pour produire un chromosome "enfant" disponible pour la prochaine itération de l'algorithme. Cette propriété de l'opérateur de mutation nous donne donc accès à l'erreur résiduelle (l'erreur pixel) relativement à chaque paramètre constituant les vecteurs C et P. Ces erreurs permettent en fait de calculer les dérivées partielles de l'erreur par rapport à chaque paramètre. Durant l'évaluation d'une génération (au cours d'une itération du GA), dès qu'un chromosome subit une mutation, l'opérateur gradient mémorise ces dérivées partielles pour construire les matrices Jaco-

bienne sans interrompre l'exécution de l'opérateur de mutation. ΔC et ΔP sont calculés de la manière suivante

$$\Delta C = -\eta \frac{J_C^T}{J_C^T J_C} e_x \tag{6}$$

$$\Delta P = -\eta \frac{J_P^T}{J_P^T J_P} e_x \tag{7}$$

où η est une valeur permettant de contrôler la différence de valeur de paramètre dans la direction du gradient.

Au début de l'évolution de l'algorithme génétique, ces matrices Jacobiennes ne sont pas pertinentes puisque l'espace de recherche du GA est très perturbé. Mais au fur et à mesure des itérations, la population produite par le GA se concentre dans une zone autour du minimum global qui est de plus en plus convexe : c'est à ce moment que l'évaluation des matrices Jacobiennes devient pertinente.

Des expérimentations ont été menées pour vérifier la stabilité des Jacobiens durant l'évolution du GA. Durant ces simulations, le GA s'exécutait normalement tandis que l'opérateur gradient évaluait la valeur du Jacobien à chaque mutation. Ces Jacobiens sont utilisés pour prédire la valeur du gène (le paramètre) compte tenu du gradient de l'erreur. Les figures 3 montrent les trois premiers paramètres des vecteur C et P évalués par le Jacobian à chaque itération. Nous pouvons constater sur ces figures que les paramètres estimés par les Jacobians deviennent stables, c'est à dire qu'ils pointent dans la bonne direction, après 7 à 10 itérations du GA; avant cela les prédictions de ces paramètres par les Jacobians sont erronées. Cela signifie qu'après un certain nombre d'itérations, la population a effectivement rejoint une zone plus stable de l'erreur autour du minimum global.

3.3 Implémentation

Cette section propose de décrire étape par étape la procédure d'optimisation hybride par algorithme génétique et descente de gradient.

1. **Initialisation.** L'image est acquise et le centre du visage évalué par un détecteur de visage. On initialise la population de N chromosomes caractérisant les vecteurs d'apparence C et de pose P.

2. **Segmentation.** Chaque chromosome spécifie une forme 3D. Chaque forme est modifiée, translatée et orientée en tenant compte des valeurs des paramètres d'apparence et de pose de chaque chromosome. La forme générée est positionnée sur l'image analysée et la texture sous-jacente est projetée dans la forme moyenne de face des visages de la base. Une normalisation photométrique est appliquée sur cette texture.

3. **Fitness.** L'erreur entre la texture normalisée précédente et celle générée par le modèle est évaluée, la somme (Eq.5) des erreurs sur tous les pixels conduit à la *fitness* associée à chaque chromosome.

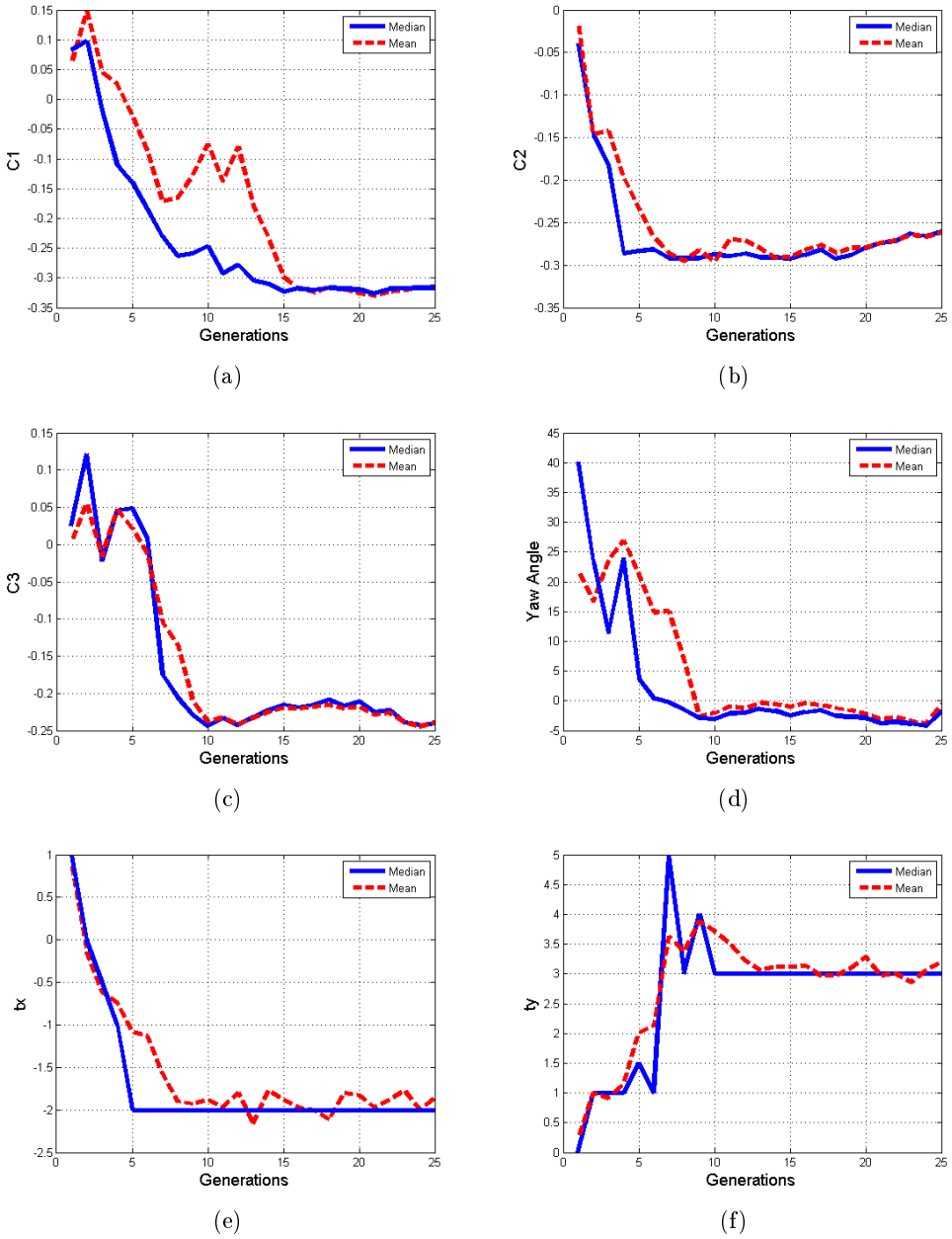


Figure 3: Evaluation de l'estimation des premiers paramètres des vecteurs C et P par les matrices Jacobienne au cours des generations de l'algorithme génétique

4. **Reproduction.** Une compétition par Tournoi permet de sélectionner une sous-population de $N/2$ chromosomes à l'intérieur de la population courante, en fonction de la *fitness* de chaque chromosome. On applique les opérateurs de crossover et de mutation sur les éléments de cette sous-population. Les parents sont remplacés par les enfants après l'opérateur de crossover. Cette nouvelle population de $N/2$ chromosomes est associée aux $N/2$ chromosomes de la population initiale n'ayant pas été sélectionnés durant le Tournoi afin de constituer une population de N chromosomes.

5. **Opérateur Gradient.** Durant la mutation de chaque gène, le vecteur d'erreur pixel est mémorisé pour calculer les matrices Jacobiennes. Comme nous l'avons expliqué précédemment, après 7 à 10 itérations, à chaque fois qu'une mutation est réalisée, la dérivée partielle de l'erreur par rapport au gène muté est évaluée. Au final, après plusieurs générations, lorsque tous les gènes ont subi une mutation, une moyenne arithmétique sur les différentes évaluations d'une même dérivée partielle dans les matrices Jacobienne est évaluée. La descente de gradient est alors appliquée sur la population de N chromosomes générée à la fin de l'opération de reproduction sur les $N/2$ premiers chromosomes, sachant que tous les chromosomes de cette population ont été ordonné en prenant en compte leur rang de Pareto.

Les étapes 2 à 5 sont itérées jusqu'à ce qu'un nombre maximum de générations soit atteint, le meilleur chromosome à chaque génération étant systématiquement conservé (stratégie élitiste). Ce nombre maximum d'itérations nous permet de comparer la complexité en termes de nombre de calcul d'erreurs de notre proposition avec d'autres techniques d'optimisations. Le chromosome ayant au final la meilleure fitness fournit les vecteurs d'apparence et de pose de la solution trouvée par l'algorithme.

3.4 Expérimentations et Résultats

Trois algorithmes d'optimisations ont été simulés à titre comparatif i) le HGOAAM (Hybrid Genetic Optimization for AAM) ii) la descente de gradient (GD) et iii) le HGA-Sim (Hybrid GA-Simplex) (Durand and Alliot [5] qui combine un Simplex et un GA et ont proposé des tests sur les fonctions classiques de Griewank et Corona). Dans les expérimentations concernant le HGOAAM et le HGA-Sim la taille de la population est de 140 chromosomes et le nombre maximum de générations pour chaque image analysée est de 25. Concernant la descente de gradient, différentes initialisations sont testées dans la mesure où le GD est très sensible aux minima locaux mais très peu consommateur de temps de calculs. Ces initialisations pavent l'espace en x et y (matrice de 3x3) autour du centre du visage détecté et sur trois valeurs de taille différentes ce qui revient à une matrice de 3x3x3 éléments correspondant à 27 initialisations différentes.

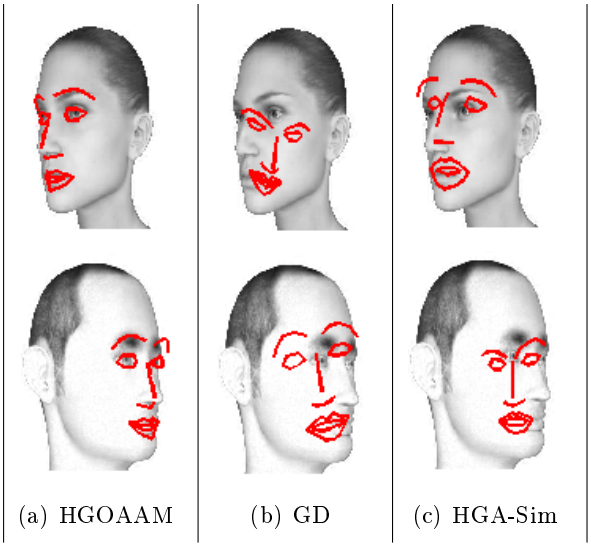
A la fin de chaque optimisation, les meilleures solutions permettent de localiser des caractéristiques telles que les yeux, le nez et la bouche. Les figures 4, 5 et 6 illustrent les performances des trois approches sur trois bases de tests différentes.

(a) HGOAAM (b) GD (c) HGA-Sim

Figure 4: Alignement des visages de la base Pointing'04 par HGOAAM, GD et HGA-Sim

(a) HGOAAM (b) GD (c) HGA-Sim

Figure 5: Alignement des visages de la base SUPELEC'08 par HGOAAM, GD et HGA-Sim



(a) HGOAAM (b) GD (c) HGA-Sim

Figure 6: Alignement des visages de la base synthétique par HGOAAM, GD et HGA-Sim

Methode	**Base**	**Images Alignées** ($GTE \leq 15\%$)	**Estimation Pose** ($E_{(\theta_{yaw})} \leq 5°$)	**Nb. de Warps**	**Temps (msec)**
HGOAAM		**41%**	—	**2500**	**83**
GD	POINTING'04 (390)	20%	—	4995	166
HGA-Sim		30%	—	**2500**	**83**
HGOAAM		**58%**	—	**2500**	**83**
GD	SUPELEC'08 (246)	47%	—	4995	166
HGA-Sim		46%	—	**2500**	**83**
HGOAAM		**46%**	**26%**	**2500**	**83**
GD	Synthetic (600)	35%	21%	4995	166
HGA-Sim		36%	13%	**2500**	**83**

Table 1: Comparaison des performances

La vérité terrain (GTE - Ground Truth Error) sur chaque image des bases de tests permet de mesurer une erreur entre les solutions trouvées par les différents algorithmes et la localisation manuelle des caractéristiques faciale sur chaque image. Un visage est considéré comme correctement aligné si la moyenne des localisations des centres de gravité des yeux, du nez et de la bouche est inférieure à 15% de la distance interoculaire du visage analysé. Etant donné la faible précision de la vérité terrain sur des images réelles concernant la valeur de l'angle θ_{yaw}, l'erreur sur ce paramètre n'a été évaluée que sur la base de visages synthétiques.

Le tableau 1 permet de comparer quantitativement les trois approches en terme de précision, de robustesse et de vitesse d'exécution. On peut voir sur ce tableau que notre proposition (HGOAAM) est bien plus efficace que les HGA-Sim et GD. Concernant la vitesse d'exécution, une descente de gradient nécessite approximativement 185 *warping* ou calcul d'erreurs par point d'initialisation, ce qui mène à un total de

4995 calculs d'erreurs étant donnée les différentes initialisations effectuées en t_x, t_y et *Scale*. HGOAAM et HGA-Sim nécessitent seulement 2500 *warping* pour localiser le visage. Chaque *warping* représente 90% du temps total d'exécution pour une itération, c'est à dire 0.03ms sur un Pentium-IV 3.2GHz. Le calcul supplémentaire induit par l'évaluation des matrices Jacobienne est négligeable, de sorte que l'alignement d'un visage par notre méthode est réalisé en 83msec sur un Pentium-IV 3.2GHz, ce qui signifie que nous pouvons exécuter notre algorithme à 12 images par seconde.

4 Système multi-vues

4.1 Multi-Objective AAM (MOAAM)

Dans un système mono-vue, une seule erreur entre le modèle et l'image analysée est minimisée. A contrario, dans un système multi-vues plusieurs erreurs doivent être minimisées en fonction du nombre de vues considérées. L'objectif est donc d'optimiser toutes les erreurs pixel $e_1, e_2, ..., e_M$ produite par les M caméras sachant que

$$e_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[I_{i,j}(C, P_j) - M_i(C)]^2} \tag{8}$$

où j varie de 1 à M et $M \geq 2$. Les vecteurs P_j caractérisent les paramètres de pose et sont liés par des offsets en rotation, scale et translation fournis lors de la phase de calibration des caméras. N est le nombre de pixel contenus dans la texture du modèle. L'optimisation de l'AAM dans un système multi vues est illustrée figure 7. Dans un tel système, le même modèle 2.5D est pris en compte simultanément au niveau de chaque caméra avec le même vecteur d'apparence C. Les vecteurs de pose de chaque caméra sont reliés par les offsets $P_{offset,j}$. Pour optimiser simultanément les M erreurs pixel, l'optimisation multi-objectifs d'un même modèle 2.5D est proposée.

Nous avons implémenté l'optimisation multi-objectifs NSGA-II (Non-dominated Sorting Genetic Algorithm) proposé par [6] pour optimiser conjointement les vecteurs d'apparence et de pose C et P. L'objectif est de trouver les valeurs de ces deux vecteurs de paramètres permettant de minimiser les erreurs pixel entre la texture générée par le modèle et celle de chacune des images acquises par les caméras. Dans ce cadre, chaque paramètre est considéré comme un gène. Tous les gènes des vecteurs C et P sont concaténés pour former un chromosome. Comme dans la section 3.1 une population de chromosomes est initialisée aléatoirement, à chaque chromosome est associé une *fitness*, une sélection de type Tournoi est mise en œuvre ainsi que des opérateurs de crossover deux points et de mutation gaussienne. La principale différence repose sur les opérateurs de sélection et de crossover qui prennent en compte le fait qu'un chromosome soit on non dominé : la fitness d'un chromosome est dépendante du rang de Pareto de ce chromosome dans la population comme c'est l'usage dans NSGA-II.

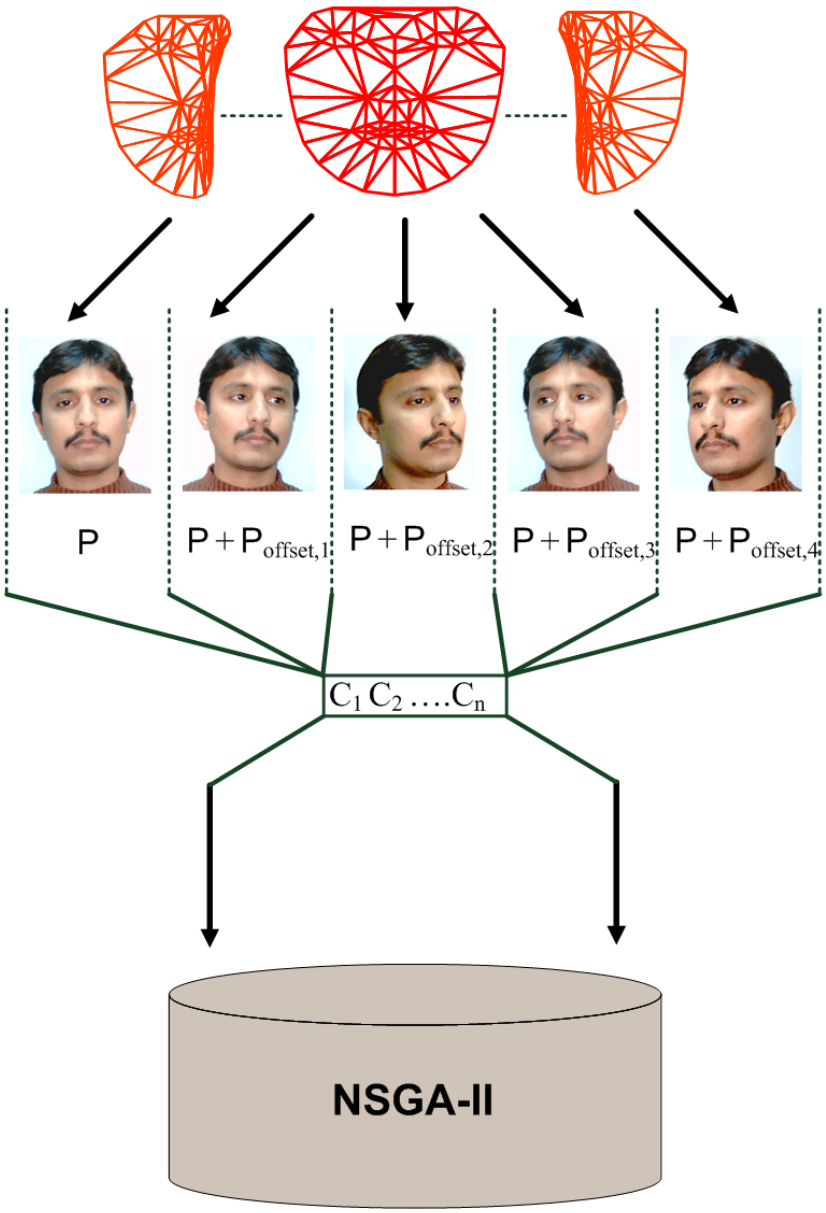


Figure 7: Un seul modèle AAM est pris en compte sur des vues différentes d'un même visage

4.2 Optimisation Multi-objectifs hybride

Dans cette section, nous présentons notre troisième contribution qui consiste en une fusion des algorithmes NSGA-II et descente de gradient pour l'alignement de visage dans un système multi-caméras.

4.2.1 Opérateur Gradient

L'évaluation des matrices Jacobienne exploitées par la descente de gradient dégrade le temps d'exécution, raison pour laquelle nous allons utiliser à nouveau l'opérateur Gradient de la section 3.2. Considérons les Jacobiens $J_{j,C}$ et $J_{j,P}$ matrices des dérivées partielles de l'erreur par rapport aux paramètres constituant les vecteurs C et P, associées à la caméra j. Au système de N caméras sont donc associés $2N$ Jacobiens. Chacun de ces Jacobiens permet d'estimer une valeur spécifique de chaque paramètre. La différence entre les systèmes mono et multi-vues réside dans la manière d'appliquer ΔC et ΔP produits par la descente de gradient. Dans le système multi-vues, ces valeurs sont prises en compte dans la deuxième phase d'optimisation du NSGA-II et avec l'aide d'un facteur spécifique appelé CIRF, relatif à la qualité de l'information associée à chaque caméra.

4.2.2 Facteur CIRF (Camera Information Relevance Factor)

Dans un système multi-caméra, l'utilisation de NSGA-II permet d'analyser l'ensemble de la population des solutions trouvées pour chaque caméra au même instant.

Considérons par exemple deux caméras installées de part et d'autre d'un écran auquel fait face un utilisateur. Selon son orientation, le visage peut faire face à trois régions R1, R2 et R3 comme indiqué dans la figure 8. Dans la région R1, les deux caméras sont aussi pertinentes l'une que l'autre en terme de qualité d'information, de sorte que les erreurs associées à l'une ou l'autre caméra ont une égale importance. Inversement si le visage est orienté vers les régions R2 ou R3, il est plus difficile pour le modèle actif de minimiser l'erreur associée à la caméra qui ne verra qu'une partie seulement du visage. Il est donc à prévoir que la population de chromosomes ait en moyenne une erreur plus élevée pour l'une des caméras comparativement à l'autre. Une vue synthétique des fronts représentés par la population des chromosomes est proposée dans la figure 8; les figures 9(a), 9(b) et 9(c) sont quand à elles des représentations réelles de ces fronts suivant la région vers laquelle pointe le visage.

Cette répartition des chromosomes suivant l'orientation des visages nous a conduit à évaluer un facteur appelé *CIRF (Camera Information Relevance Factor)* qui traduit la pertinence d'une caméra par rapport aux autres sachant qu'une caméra sera d'autant plus utile pour l'optimisation que le visage lui fera face. Ce facteur sera ensuite exploité dans la descente de gradient.

Considérons qu'à chaque instant, le visage est orienté de telle façon que l'information issue de chaque caméra est pertinente mais que le taux de pertinence varie selon la caméra. Pour évaluer ce taux, nous proposons une technique pour analyser de façon automatique la population classée par l'approche de Pareto. Cette procédure démarre

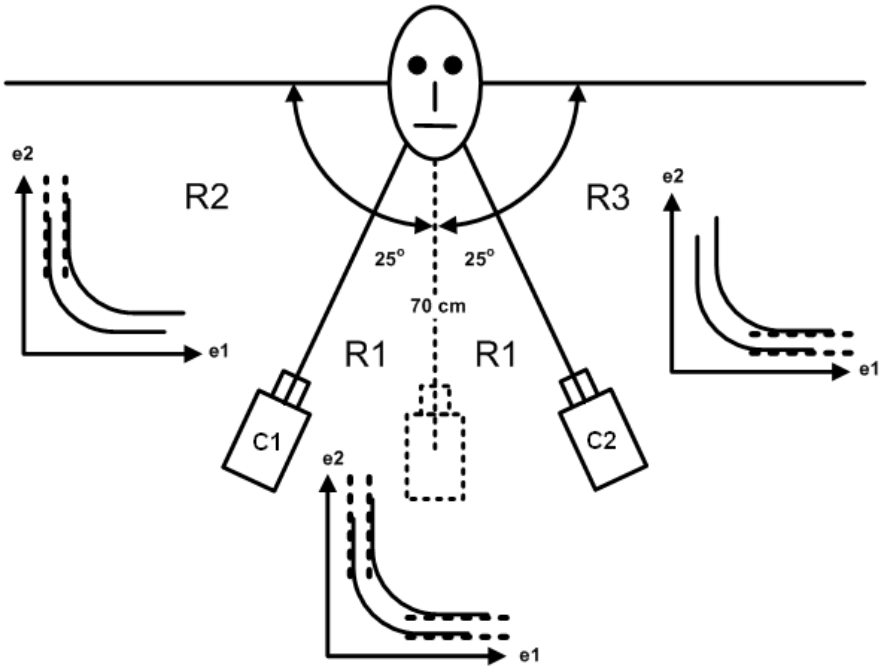


Figure 8: Système multi vues

lorsque le critère d'arrêt associé à la première phase d'optimisation du NSGA-II est atteint pour une image donnée. Il consiste à i) calculer la valeur médiane de chaque erreur pixel $e_{1,i}$ et $e_{2,i}$ associée à chacune des caméras des N chromosomes de la population, ii) tirer une ligne entre cette valeur médiane de l'erreur (vecteur à deux composantes puisque nous avons deux caméras) et l'erreur minimale atteinte par les deux caméras (vecteur d'erreur à deux composantes $[e_{min}\ e_{min}]^t$ voir ci-après la définition de e_{min}). Ces lignes sont illustrées sur les figures 9(a), 9(b) et 9(c). A partir de ces valeurs, les CIRF associés à chaque caméra peuvent être évalués de la manière suivante.

$$\psi = \frac{arctan\left(\frac{\tilde{e_2}-e_{min}}{\tilde{e_1}-e_{min}}\right)}{\pi/2} \tag{9}$$

où

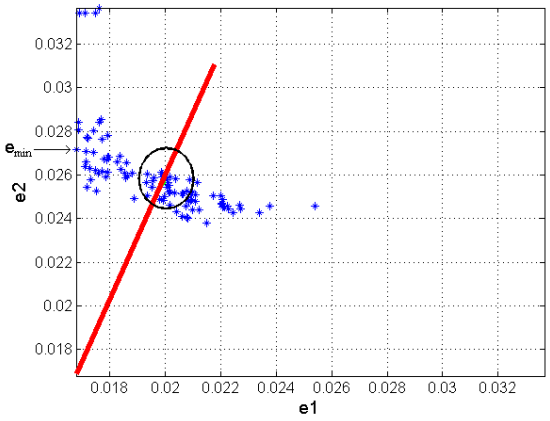
$$\psi = Camera\ Information\ Relevance\ Factor,$$

$$\tilde{e_1} = Median\ de\ e_{1,i} \quad 0 \leq i \leq N,$$

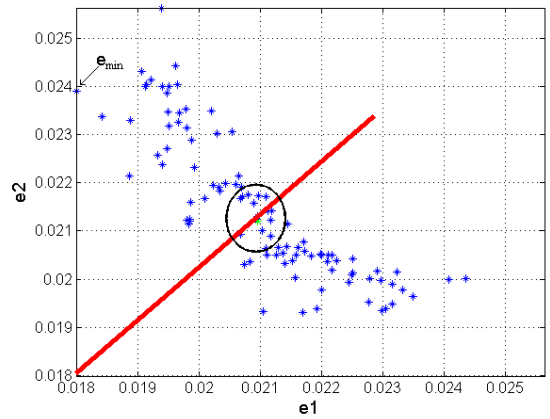
$$\tilde{e_2} = Median\ de\ e_{2,i} \quad 0 \leq i \leq N,$$

$$e_{min} = Minimum\ (Minimum(e_{1,i})\ ,\ Minimum(e_{2,i}))$$

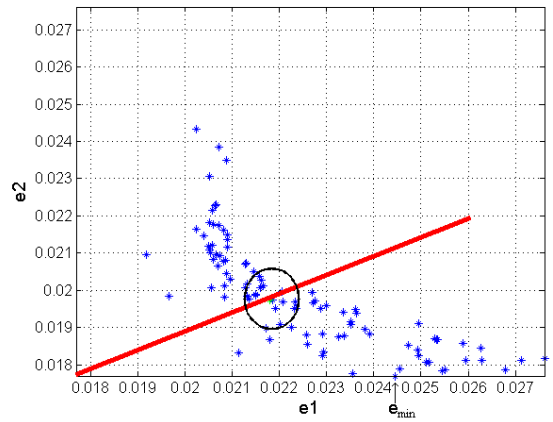
La caméra 1 des figures 9(a), 9(b) et 9(c) a un CIRF de 0.79, 0.52 et 0.30 respectivement, ce qui signifie que les caméras les plus pertinentes sont la caméra 1, les deux caméras et la caméra 2. La valeur du CIRF varie entre 0 et 1. Si un visage est orienté de manière à ce qu'une caméra ne voie pas l'un des profils, la valeur du CIRF associé à cette



(a) $\theta_{yaw} \in R2$



(b) $\theta_{yaw} \in R1$



(c) $\theta_{yaw} \in R3$

Figure 9: Erreurs associées à chaque chromosome selon l'orientation du visage

caméra doit être de zéro. Cependant, étant donnée la diversité des solutions proposées par NSGA-II, le CIRF n'atteint jamais les valeurs extrêmes de 0 ou 1 même s'il s'en approche. Nous pouvons donc utiliser la valeur du CIRF pour négliger l'information de l'une ou l'autre caméra lorsqu'elle n'est pas pertinente. La valeur du CIRF est utilisée à titre de coefficient de pondération pour générer une erreur global e_{total} minimisée par la descente de gradient : on se ramène donc à un problème mono-objectif.

$$e_{total} = e_1 * \psi + e_2 * (1 - \psi) \tag{10}$$

La descente de gradient minimise donc une seule erreur, fruit de la pondération des erreurs associées à chacune des caméras suivant leur pertinence.

Il nous faut à présent sélectionner un sous-ensemble de la population de chromosomes sur lequel va être appliqué la descente de gradient. Le critère de sélection des solutions proposé dans les algorithmes évolutionnaires privilégie la survie des chromosomes ayant la meilleure *fitness*, qu'il s'agisse d'une optimisation mono ou multi-objectifs : c'est ce que nous allons reproduire ici en prenant en compte le CIRF. Nous utilisons la droite définie précédemment (voir Fig.9) pour classer par ordre croissant les chromosomes en fonction de leur proximité par rapport à cette droite. Un nombre spécifique des chromosomes les plus proches de la droite est sélectionné, une descente de gradient est appliquée sur chacun d'entre eux. En règle générale, ils sont positionné dans le voisinage de l'erreur médiane (régions encerclées dans les figures 9(a), 9(b) et 9(c))

4.3 Implémentation

Cette section décrit étape par étape l'alignement de l'AAM 2.5D optimisé par l'algorithme hybride multi-objectifs.

1. **Initialisation.**

   Les images sont acquises à partir des M caméras avec le centre de gravité du visage inconnu localisé dans chaque image par un détecteur de visage. On initialise de façon aléatoire la population de N chromosomes caractérisant les vecteurs d'apparence C et de pose P. La *fitness* des chromosomes est évaluée à partir de l'équation 8 par rapport à chaque caméra. A un chromosome correspond donc un ensemble de valeurs $e_1, e_2, ..., e_M$. Chacun des chromosomes est alors classé selon son rang de Pareto.

2. **Reproduction.** Une sélection par Tournoi permet de sélectionner une population de N chromosomes "Parents". Dans cette nouvelle sélection il peut donc arriver qu'un même chromosome soit représenté plusieurs fois. Les opérateurs de crossover et de mutation sont appliqués et conduisent à une population $Q[t]$ de N chromosomes "Enfants". Les "Parents" et les "Enfants" constituent une nouvelle population de $2N$ chromosomes, à nouveau classés selon leur rang de Pareto.

3. **Segmentation.** Chaque chromosome correspond à une forme 3D de visage. Cette forme est positionnée dans chacune des M images et prend en compte les offsets

relatifs à la caméra associée. Par exemple pour l'image j, les offsets $\theta^{offset}_{(aw,j)}$, $\theta^{offset}_{(pitch,j)}$, $\theta^{offset}_{(roll,j)}$, $tx^{offset}_{(j)}$, $ty^{offset}_{(j)}$ et $Scale^{offset}_{(j)}$ de la caméra j seront évalués relativement à la caméra centrale. Les paramètres C quand à eux restent identiques pour chaque vue. La texture de l'image contenue dans la forme positionnée est *warpée* dans la forme moyenne définie dans la phase de modélisation. Cette texture warpée est normalisée en photométrie. Les offsets introduits précédemment sont relatifs à la caméra centrale ce qui signifie que la forme générée par un chromosome sans offset représente un visage vue par la caméra centrale. Pour cette raison, nous évaluerons expérimentalement notre algorithme sur la base de cette image frontale.

4. **Fitness.** Les erreurs pixels $e_1, e_2, ..., e_M$ sont alors évaluées (voir Eq.8) entre la texture normalisée de chaque image et celle générée par le modèle et cela pour chaque chromosome.

5. **Opérateur Gradient.** Durant la mutation de chaque gène dans la phase de reproduction, l'image de l'erreur est mémorisée pour le calcul futur des matrices Jacobiennes. Dès qu'une mutation est réalisée, une dérivée partielle de l'erreur en fonction du gène muté est calculée. Au final, lorsque tous les gènes ont subit cet opérateur de mutation, une moyenne arithmétique est effectuée sur chaque dérivée partielle (compte tenu du fait qu'un même gène a subit plusieurs mutations) et les matrices Jacobienne sont alors disponibles pour la descente de gradient.

6. **Classement de Pareto** Un classement des chromosomes selon Pareto est effectué sur l'ensemble de la population pour produire des fronts de Pareto. Tous les chromosomes sont ainsi classés selon leur rang correspondant au numéro du front de Pareto auquel ils appartiennent. C'est grâce à ce classement par rang que nous sommes capables de comparer les chromosomes les uns par rapport aux autres dans un contexte multi-objectifs. Les chromosomes de même rang se distinguent par leur distance de *crowding* qui prend en compte pour un chromosome le nombre de chromosomes qu'il possède dans son voisinage.

7. **Sélection.** Dans la phase de reproduction, la taille de la population devient le double de la population originale, c'est à dire $2N$. Pour maintenir une population de taille constante, seulement N chromosomes (ayant les rangs les plus faibles) sont conservés. Cependant, si les chromosomes du dernier rang considéré mènent à une population supérieure à N, alors on ne sélectionne que ceux ayant une distance de crowding élevée pour maintenir la diversité sur chaque front.

Les étapes de 2 à 7 sont répétées jusqu'à ce qu'un nombre maximum de générations soit atteint, le meilleur chromosome à chaque génération étant systématiquement conservé. Comme indiqué dans le système mono-vue, ce nombre maximum de générations permettra de comparer la complexité algorithmique des différentes approches testées. La répartition des chromosomes selon les différents fronts de Pareto est analysée afin de calculer le CIRF de chaque caméra et de sélectionner les chromosomes sur lesquels sera

appliquée la descente de gradient comme présentée dans la section 4.2.2. Au final, les matrices Jacobienne évaluées durant les générations sont utilisées dans les descentes de gradient et conduisent à la production d'un meilleur chromosome.

Nous avons présenté un algorithme d'optimisation avec M caméras. Que M soit supérieur ou égal à deux, le principe reste le même. Le nombre de matrices Jacobienne augmentera linéairement avec le nombre de vues considérées et les fronts de Pareto seront représentés dans des espaces à M dimensions. Dans la section suivante, nous testerons l'algorithme proposé dans un système avec deux caméras seulement.

4.4 Experimentations et Résultats

4.4.1 Experimentations

Trois tests ont été menés.

AAM Mono-objectif (SOAAM Single-Objective AAM). C'est la caméra centrale qui est utilisée, un AAM 2.5D optimisé par un algorithme génétique classique permet d'aligner le visage.

AAM Multi-objectif (MOAAM - Multi-Objective AAM). Les deux caméras latérales permettent d'aligner le visage avec un AAM 2.5D optimisé par un NSGA-II.

AAM Multi-Objectif Hybride (HMOAAM - Hybrid Multi-Objective AAM). C'est un NSGA-II optimisé avec l'algorithme hybride présenté dans la section 4.3 qui est utilisé dans un système à deux caméras.

4.4.2 Résultats

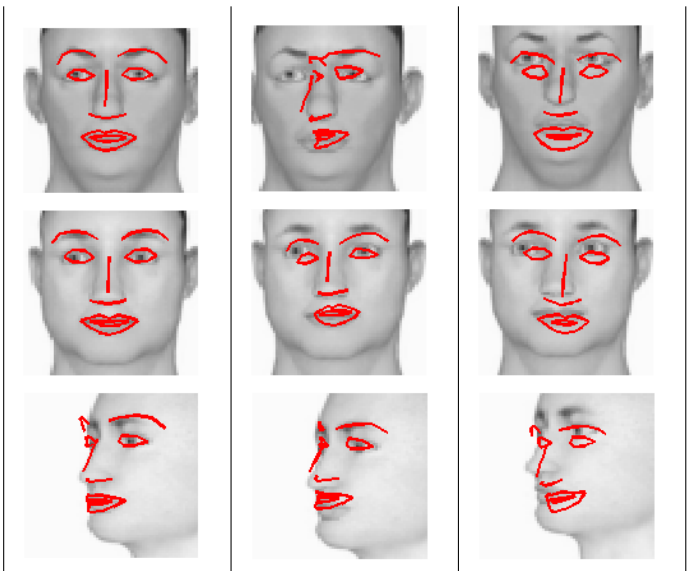
Les meilleurs chromosomes (par rapport aux erreurs) obtenus par HMOAAM, MOAAM et SOAAM donnent la texture et la forme du visage trouvé. On en extrait les caractéristiques telles que les yeux, le nez et la bouche. Ces localisations en terme de contours sont présentées dans la figure 10 en ce qui concerne les images réelles acquises par les caméras et dans la figure 11 pour la base synthétique.

Le pourcentage de visages correctement alignés a été évalué en prenant en compte la vérité terrain (GTE - Ground Truth Error) qui donne le centre de gravité des yeux, du nez et de la bouche. Nous suivons le même protocole qu'en 3.4 : un visage est considéré comme correctement aligné si la moyenne des localisations de centres de gravité est inférieure à 15% de la distance interoculaire du visage analysé. De la même manière la pose d'un visage synthétique sera considérée comme correcte si l'angle θ_{yaw} détecté est correcte avec une précision inférieure à 5°.

On peut constater dans le tableau 2 que les HMOAAM sont bien plus efficaces que les MOAAM et SOAAM. Concernant le temps d'exécution, les HMOAAM nécessitent 3000 *warping* pour aligner un visage orienté. Sachant qu'un seul warping prend 0.03ms sur un Pentium-IV 3.2GHz et qu'il représente à lui seul 90% du temps de traitement d'une itération, les HMOAAM fonctionnent a 10Hz pour un temps d'exécution de 100ms. Pour pouvoir comparer les performances des trois approches, les deux autres algorithmes ont été paramétrés afin que leur temps d'exécution soient du même ordre. Le SOAAM a deux fois plus d'itérations que les deux autres algorithmes, dans la mesure où il n'exploite

(a) HMOAAM (b) MOAAM (c) SOAAM

Figure 10: Alignement de visages réels par HMOAAM, MOAAM et SOAAM



(a) HMOAAM (b) MOAAM (c) SOAAM

Figure 11: Alignement de visages synthétiques par HMOAAM, MOAAM et SOAAM

Table 2: Comparaison des performances des HMOAAM, MOAAM et SOAAM

Method	Database (Images)	Images Aligned ($GTE \leq 15\%$)	Pose Estimated ($E_{\theta_{yaw}} \leq 5°$)	pop x gen	No. of warps	Time (msec)
HMOAAM	Synthetic (4160)	**48%**	**42%**	100 x 14	3000	100
MOAAM	Synthetic (4160)	39%	37%	100 x 15	3000	100
SOAAM	Synthetic (4160)	22%	17%	100 x 30	3000	100
HMOAAM	Webcam (246)	**61%**	—	100 x 14	3000	100
MOAAM	Webcam (246)	48%	—	100 x 15	3000	100
SOAAM	Webcam (246)	43%	—	100 x 30	3000	100

qu'une seule vue. Le HMOAAM quand à lui n'aura que 14 itérations comparées au 15 itérations du MOAAM pour compenser le coût algorithmique de la descente de gradient dans la seconde phase de l'optimisation.

5 Conclusions

Dans ce rapport de thèse, nous avons proposé des solutions pour aligner un visage (détecter ses traits caractéristiques) inconnu devant une ou plusieurs caméras, en particulier lorsque ce visage produit de forts mouvements latéraux.

Notre première contribution concerne le modèle actif d'apparence 2.5D qui permet de *warper* une simple vue de face d'une personne sur un modèle 3D de son visage.

Notre seconde contribution a consisté à proposer une optimisation hybride basée sur un algorithme génétique et une descente de gradient pour un système d'acquisition mono-vue. La descente de gradient est intégrée dans un opérateur spécifique qui travaille de concert avec l'opérateur de mutation. L'intérêt est que de la sorte, le coût algorithmique est dérisoire et conduit à un algorithme d'optimisation à la fois rapide, robuste et efficace. Les performances d'une telle approche ont été illustrées sur plusieurs bases réelles et synthétiques, ces tests ont permis de quantifier notre apport comparé à des techniques plus classiques.

Enfin notre dernière contribution tient dans la proposition d'un algorithme d'optimisation hybride dans un système multi-caméra nécessitant une optimisation multi-objectifs. Nous avons proposé une méthode automatique pour qualifier la pertinence des informations fournies par chacune des caméras afin de les prendre en compte de façon appropriée lors de la descente de gradient qui est intégrée à l'algorithme d'optimisation NSGA-II. Des comparaisons quantitatives et qualitatives avec d'autres approches mono et multi-objectifs montrent l'intérêt de notre méthode lorsqu'il s'agit d'évaluer la pose et les traits caractéristiques d'un visage inconnu.

Notons que ce travail de thèse s'est focalisé sur l'estimation de la pose et l'alignement d'un visage inconnu sans connaissances a priori; en mode de suivi, il est possible de

réduire de façon importante le temps de traitement en exploitant les informations recueillies dans les images précédemment traitées du visage. Nos propositions prennent alors tous leur sens dans une application temps réel où le problème de la robustesse dans le suivi de visage est essentiel.

Notre système peut donc être utilisé comme *tracker* de visage mais pas seulement. Dans le domaine de la biométrie par exemple, lorsqu'il s'agit de reconnaître un visage ou de l'authentifier par rapport à une photo d'identité, il est nécessaire d'aligner très correctement le visage (détecter très précisément le centre des yeux, du nez et de la bouche) et d'être capable de détecter sa pose afin de synthétiser une image frontal normalisée et correctement cadrée sur laquelle s'appuieront les algorithmes de reconnaissance de visage à proprement parler.

Chapter 1

Introduction

Contents

1.1 Cognitive Radio

Our research team, SCEE works in the domain of Cognitive Radio Systems. A Cognitive Radio approach proposed by Mitola [7] extends the concept of a hardware radio and a software defined radio (SDR) in a radio that senses and reacts autonomously to its operating environment changes. A Cognitive Radio (CR) equipment is a radio device that supports the smart facilities offered by future cognitive networks. In future several categories of equipments will exist, depending on their processing capabilities; it means

that apart from the usual radio signal processing elements, these equipments also have to integrate a set of new sensing capabilities for the CR support. The particularity of a CR equipment is to integrate decision and sensing capabilities. All this makes a CR equipment aware of its environment and permits to adapt its behavior to the context. Context and its sensing should be considered here at a large scale. All information that can help the radio to better adapt its functionality for a given service in a given environment, in other words under given constraints, is worth being taking into account. Then as we make no restriction on the sensor's nature, it is possible to draw the general approach exposed in figure 1.1. Sensors of a CR equipment are classified in function of the OSI layers they correspond to, with a rough division into three layers.

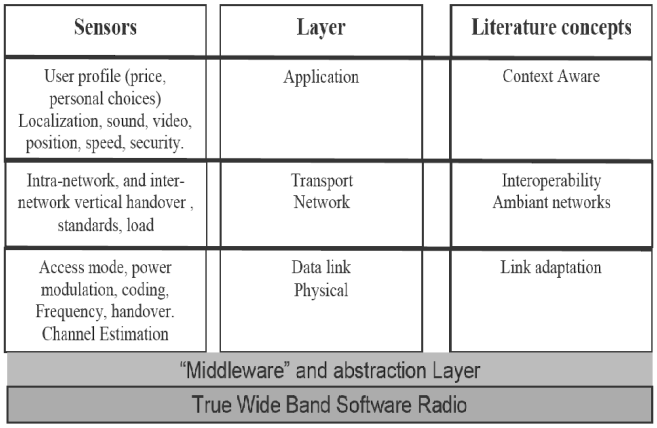


Figure 1.1: Cognitive Radio OSI Model

Lower layers of the OSI model corresponds to all of the sensing information related to the physical layer: propagation, power consumption, coding scheme, etc. At the intermediate level are all information that participate to vertical handovers, or can help to make a standard choice, as a standard detection sensor for instance. It also includes the policies concerning the vicinity, the town or the country. The CR sensors or the highest layer are especially related to the applications and all that concerns the human interaction with the communicating device. It includes the user's habits, preferences, policies, profile etc. If a user has the habit to connect to a video on demand (VoD) service every evening while coming back home from office by metro, a CR terminal should be aware of it to plan all the requirements in terms of battery life, sufficient quantity of credit on his contract and vertical handover succession depending on each area during the trip. The equipment can be aware of its environment with the help of sensors like microphone, video-camera, bio-sensors, etc. As CR technology is at its early stage, it is difficult to foresee all the possibilities. One can think that user's bio metric information and/or facial recognition will ensure user identity and equipment security. Video-camera could also be used to indicate if the terminal is outside or inside a building. Another example is given in the context of video conferencing, a separation between the face of the speaker and the background could help decreasing the data rate

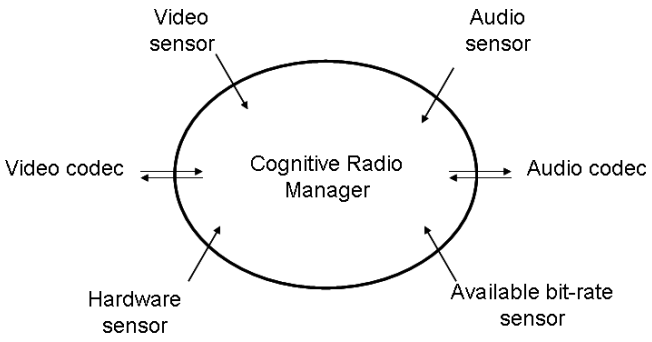


Figure 1.2: Cognitive Radio Manager

while slowly refreshing the background of the image. Similarly this face separation also helps to choose different image/video compression techniques suitable to the application. Finally, video and audio sensors can produce useful information about the content of a streaming video. These informations will guide the choice of the most adapted codec which must be used in order to optimize the quality of the display taking into account the global context of the radio transmission. These codecs can perform lossy and/or loss-less compression of the objects, displayed. In verbal human communication, the most vital part of human is his face, whose analysis is essential in CR. In the next section we present the importance of facial analysis in a CR equipment.

1.1.1 Facial Analysis in CR

As discussed in the previous section, video sensors play an important role in a human machine (CR equipment) interaction. One of the application of these sensors is to grab the facial information of the user. Human face and its facial features (e.g. eyes, nose, mouth and eyebrows) are actually the reflection of its inner emotional state and personality. They are also believed to play an important role in social interactions, as they give clues to the state of mind and therefore help the communication partner to sense the tone of a speech, or the meaning of a particular behavior. For these reasons, human face can be identified as an essential non-verbal communication channel.

In a CR equipment, Cognitive Radio Manager (CRM) adapts the radio equipment parameters taking into account sensor's information to provide the best functionality with in the available resources. According to the sensor's information, the CRM will define the optimal configuration to give the best service. As illustrated in figure 1.2, the audio and video sensors will be considered by the CRM to parameterize the codecs.

For example for an audio coding, if a speech signal is not detected a general audio coding like a TwinVQ (transform-domain weighted interleaved vector quantization) is applied. On the other hand, when a voice is detected, a codec dedicated to speech compression like a Code Excited Linear Predictive (CELP) can be used to increase the quality of the delivered signal. In this way, for the same bit rate the subjective quality of a voice compressed by a speech coding is better than other audio signal compressed

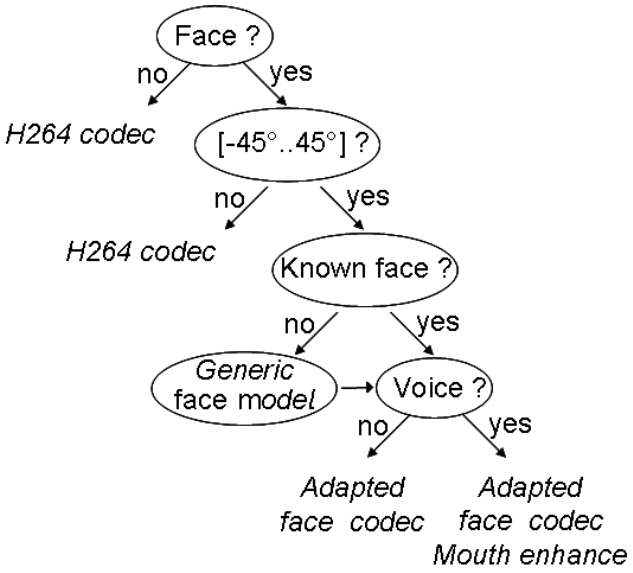


Figure 1.3: Video Sensor

by general coding.

Similarly in video coding, according to the presence or absence of a face, a generic H264 codec or a face codec is used. Figure 1.3 represents the decision tree used by the CRM to specify the image codec configuration knowing the state of sensors. The bubbles characterize the sensors information, the tree's leaves give the decision for a video codec configuration. This decision tree is explained by dividing it in terms of sensors information as given below.

- *Face*: The first input sensor inform the CRM about the presence of a face. In the case of absence of a face, a generic video codec (H264) is used on the whole image. Whereas, presence of a face requires facial analysis by the subsequent sensors.

- *Facial Pose*: If a face is detected then the information about its approximate pose is taken into account without its identity. If the person is not really facing the camera (face pose angle larger than semi-profile one) then it means that the most relevant information is not in the face itself: the H264 codec is used because the background information is for sure important. If the person is facing the camera then it is interesting to know if the face is already known in the system or not.

- *Facial Identity*: If the face is unknown, then a generic configuration of the facial analysis is required. It can learn and analyse the face by the varying a diverse facial model, obtained from a combination of various facial classes. On the other hand, if the face is well known by the system then its model is already available. The purpose of learning and analysing the facial identity is to precisely estimate facial pose and location of facial features.

- *Facial Features*: At the end, if the voice signal is detected then the mouth region is important to capture and must be enhanced comparatively to the eyes, skin and of course to the background. In that case, the background must be highly compressed since it does not a priori carry relevant information for the communication. If a voice signal is not detected, then the most important information is in the eyes which must be enhanced relatively to the others face characteristics (nose, mouth, skin) knowing that the background can be highly compressed.

Video sensors are proved to be significant part of a CR equipment whether they are installed on a computer or on a mobile phone. In the presence of a face, the choice of using a generic video codec (H264) or a specialized codec which provide high compression is based on the application of face analysis embedded in the CRM. In the next section bottlenecks in the domain of facial analysis are discussed.

1.1.2 Problem Statement

The inputs represented in figure 1.3 which decide the image codec configuration are: 1) Voice detection, 2) Face detection, 3) Face orientations, 4) Face identity (Known / Unknown) and 5) Facial features detection. Solutions to the first two sensors are currently implemented in the market equipment. There is no obvious solutions to fulfill the last three sensors. Problem encountered in the solutions of these sensors are explained in details in the subsequent sections.

Face Orientations: Facial pose estimation is one of the necessity of the facial analysis in a CR equipment. As shown in the figure 1.3 facial pose estimation is the second most important step required after the detection of the face, due to the fact that the user is not obliged to remain in frontal view in front of the camera. For a robust interactive system it is necessary to implement an algorithm which can estimate facial pose of a face with in-plane (first row of figure 1.4) and out-of-plane (second and third rows of figure 1.4) rotation.

Unknown Faces: Classification of a human race is based on race, age, skin color, facial features etc. Various classes of human exist in the world. If the face belongs to the database, it is possible to recognize the face, thus it becomes a face recognition application. On the contrary if the face is unknown then the system should be able to extract the facial information of that unknown face. It can be accomplished by approximating the user's face by varying the appearance of a generic human face class. Since face recognition is very sensitive to the alignment of the eyes, nose and mouth therefore it becomes necessary for the system to accurately extract the required facial information of an unknown face. The scenario of unknown face can be explained with an example in which an unknown person enters a secured area by passing through security office where his photographic identity is used to update the database by the system. In this thesis a solution is presented for the analysis of unknown faces.

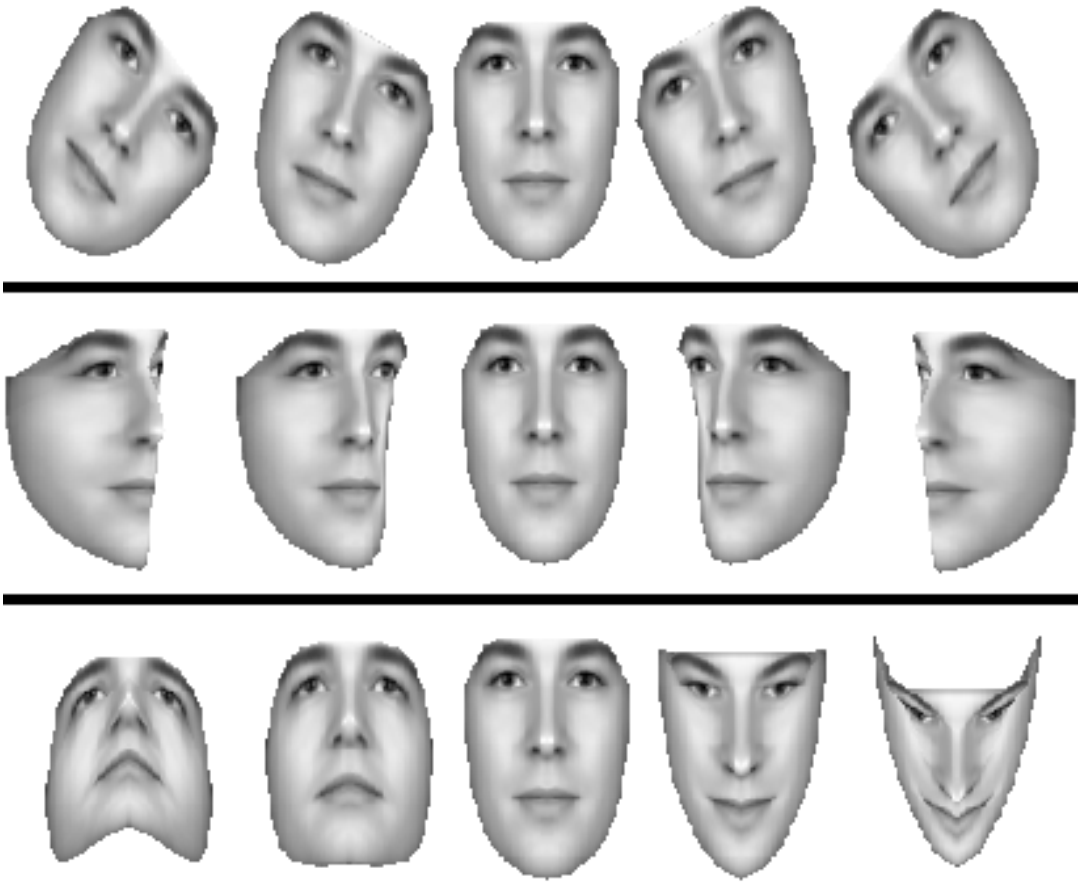


Figure 1.4: Roll (top), Yaw (center) and Pitch (bottom) of a face

Facial features detection: Detection of face characteristics like eyes, nose and mouth is again one of the necessity of the facial analysis in a CR equipment. As discussed in section 1.1.1 in the presence of a voice, mouth region is important and must be enhanced comparatively to the eyes, skin and background. Similarly blinking of the eyes gives the partner a realistic look as well as a way of expressing a thought. However locating these features in frontal view faces is not that difficult, whereas in faces making lateral movements it is one of the most challenging task, due to the occlusion/deformation of these features in lateral movement. For example in a semi profile view one of the eye of the face become smaller compared to the other. Similarly one half of the mouth appears bigger than the other half as shown in the central row of figure 1.4.

Each problem encountered in the face analysis system, discussed in the previous sections, is even difficult to be dealt with alone and their combination makes the system more difficult. For example pose estimation of a face moved laterally in a semi profile view is one of the difficulty and if the face is unknown as well, it adds to the difficulty of the face analysis system. Additionally facial features localization in this scenario requires a methodology, robust and efficient enough to tackle all these problems. Therefore the problem statement could be given as *"Pose estimation and facial features localization of unknown and oriented faces"*. Next sections presents the thesis's proposed solutions in this domain.

1.2 Face Analysis Solutions

Various method have been proposed in the domain of face alignment and its features detections. These methods can be grouped in two main categories; Pixel-based and Model-based methods. Pixel-based methods are briefly explained with some references in section 1.2.1 along with the explanation of their imperfection for solving the problems stated in this thesis. A brief history of model-based methods is given in section 1.2.2 and their detailed description along with references is given in the next chapter.

1.2.1 Pixel-Based Methods

In pixel based methods (a.k.a image-based methods) facial features are detected from the whole (holistic approaches) or various parts (non-holistic approaches) of the facial images taking into account their pixel intensities. These methods extract features from images without relying on extensive knowledge about the object of interest. They have the advantage of being typically fast and simple. However, pixel-based becomes unreliable and unwieldy, when there are many different views of the same object that must be considered. In these methods sometimes specific image filters are applied on the images to enhance the salient features of the facial image. In some of these methods the color or gray level intensity of the pixel is used for the segmentation of the facial features. Segmentation is also performed by initializing a small rectangular, circular or elliptical shape around a particular feature taking into account again the pixel intensity

of the selected region. Some of the well-known methods under this category are given in table 1.1. These methods are divided in two categories.

Pixel-Based Methods
Template matching [8][9][10]
Ellipse/rectangle fitting [10][8]
Skin color segmentation [8][11][12]
Edge-like blob map [11, 12]
Luminance distribution [13]
Gabor Wavelet [14][15][16]
Neural Networks [16][17]
Separability filter [18][9]

Table 1.1: Pixel-Based Methods

1.2.1.1 Methods with Preprocesses

Some methods use pre-process on the facial images to enhance its salient features. These pre-processes include skin color/luminance segmentation, edge-like blob map, Gaussian derivative and separability filters etc. For decision algorithms they usually use template matching either on whole face or on specific facial features.

Vezhnevets et al. [8] describes algorithms for face and facial features detection in still frontal images. First, facial area is detected using skin color segmentation and adaptive ellipse fitting. Next, eye positions are estimated by finding eye-shaped and eye-sized areas of red channel sharp changes. Finally, exact facial contours of eyes, eyebrows, nose, mouth, chin, and cheek are estimated by employing deformable models, template matching, and color segmentation. Kawaguchi and Rizon [9] presented an algorithm which first detects the face region in the image and then extracts intensity valleys from the face region. Next, the algorithm extracts iris candidates from the valleys using Hough transform, separability filter and template matching. Lee et al. [11, 12] proposed a facial feature detection method based on local image area and direct pixel-intensity distributions, in which they proposed two novel concepts; the directional template for evaluating intensity distributions and the edge-like blob map with multiple strength intensity. Final candidate face region is determined by both obtained locations of facial features and weighted correlations with stored facial templates. In case of color image, faster detection of both facial features and face is feasible by using the chromatic property of facial color. Gourier et al. [18] proposed a method in which facial images are represented by a vector of scale normalized Gaussian derivatives at each pixel. Gaussian derivatives provide a feature vector for local appearance at each pixel. These vectors forms clouds of points in the feature space. K-means clustering is used to determine a cluster that provide a detection of salient facial features.

1.2.1.2 Decision Algorithm

Depending upon the decision algorithm some methods analyse whole face while some of them concentrate on the specific facial features like eye, nose and mouth etc.

Whole Face Duffner and Garcia [17] presented a technique for the face alignment using Convolution Neural Networks. Their CNN based method is trained to output the transformation parameters corresponding to a given mis-aligned face image. They aligned the face simultaneously with respect to x/y translation, scale and in-plane rotation. Similarly Stathopoulou and Tsihrintzis [19] also used artificial Neural Networks for the training of different facial expressions and extracted facial features of a query face in the testing phase. Praseeda and Sasikumar [16] present a method to analyze facial expression from images by applying Gabor wavelet transform (GWT) and Discrete Cosine Transform (DCT) on face images. Radial Basis Function (RBF) based Neural Network is used to classify the facial expressions. Kruger et al. [14], R. S. Feris and Jr. [15] describes a method in which faces are detected and tracked in a video sequence using Gabor wavelet networks. This process also allows locating and extracting facial feature regions around the eyes, nose and mouth.

Specific Feature Lu and Yang [10] proposed a new method for eye detection based on rectangle features and pixel-pattern-based texture feature (PPBTF). First, Adaboost cascade classifier by rectangle features is constructed to do rough eye detection in a front facial image. Second, the result image patches are cropped and scaled to 24x12 to compute the features of PPBTF, then, put these features into an Adaboost and SVM classifier for an accurate detection. Wu and Zhou [13] proposed a method in which after face detection eye-analogue segments at a given scale are discovered by finding regions which are roughly as large as real eyes and are darker than their neighborhoods.

The appearance of the face depends on the angle at which a given face is being observed. Pose variations occur due to the in-plane and out-of-plane rotations of faces. Especially out-of-plane rotations are difficult to handle in these methods. These methods are usually used for locating the approximate position of facial features. Therefore they are not robust to out-of-plane rotations of a face and facial image background. Most of the research work discussed above is for frontal view facial images.

1.2.2 Model Based Methods

The facial structure can also be described with the help of 2D or 3D deformable face models. In the year 1993 and 1994 the deformable face model-based methods (a.k.a Geometric based methods) were introduced as PDM by Sozou et al. [20] and as ASM by Cootes and Taylor [21]. Wiskott et al. [22] proposed Elastic Bunch Graph Matching where they mapped a deformable grid onto a face image which is comprised of nodes of feature graphs. Each node consists of Gabor jets, which are filter response Gabor wavelet extracted at a given image point. These modeling methods only contain the facial shape variability and do not contain the variability of complete facial textures.

Textural information of the face plays an important role in the segmentation of the facial features as one can see there exists a category of methods which employ pixel intensities of various regions of a face (see section 1.2.1). Realizing the importance of texture in a facial image the development of model-based methods was extended to another family of deformable models called FAM by Lanitis et al. [23] and AAM by Cootes et al. [1]. Among them AAM is the most well known method for the facial analysis systems. Later, Blanz and Vetter [24] and Ahlberg [25] extended these texture based deformable model to three dimensions.

Model-Based Methods
Active Shape Model [21]
Point Distribution Model [20]
Flexible Appearance Model [23]
Elastic Bunch Graph Matching [22]
Active Appearance Model [1]
3D Morphable Model [24]
Candide 3D Face Model [26][25]

Table 1.2: Model-Based Methods

Different modes of creating deformable facial models exists, depending upon the shape model, shape/texture model and variations incorporated in them. Shape models can be build either by placing few key landmarks points around significant facial features or placing a grid of equidistant points on the whole face. Similarly shape and texture models variations could be combined to obtain appearance parameters or they can be handled separately in the segmentation phase. Various shape and/or texture based deformable facial models being used by the community are shown in table 1.2. All of these deformable face models methods are explained in detail in the section 2.1 of the next chapter.

We need a method which would be more robust to the deformations of a face, so that facial features and pose of an unknown face could be extracted more efficiently. For this kind of application model-based methods are the most suitable approaches. They are more robust compared to pixel-based methods in facial pose estimation, features detection, expression detection, analysis of unknown faces, image background, illumination variations and occlusions (beard/mustache). Out-of-plane facial rotations can be addressed by warping techniques, in which the face model can warp the oriented face into frontal view. The only drawback of these methods is their requirement of high computational time, which is dealt with efficient optimizations technique to make them time efficient. This thesis focuses on such methods and a detailed state of the art dedicated to these methods is presented in the next chapter.

1.3 Hardware Solutions

Facial information, required for facial analysis by a CR equipment, is captured by video sensors. There are many types of video sensors, but the most common and easily available one is a digital camera in the form of a video camera, still camera, webcams or mobile phone cameras. Cameras are input devices of a machine compatible to human eyes. Most of the time researchers do not concentrate on this topic and take the available facial image databases made from single camera. In this thesis the use of a single and/or multiple cameras is addressed in detail.

1.3.1 Single Camera

Single camera approach is the most simplest approach to acquire images. Although cameras calibration is required, however, only intrinsic parameters are calibrated. Most of the work in face analysis domain has been done by using a single camera configuration.

1.3.2 Double Camera

The use of double camera is to replicate human eyes. Single camera can only give the XY coordinate of the object in the image and if the coordinates of the camera are known it can precisely localize the object with respect to the global coordinates. On the other hand stereo camera can also give the depth of the object, provided that the images from each camera are temporally synchronized. Various methods of stereo vision (epipolar geometry and triangulation) are used to register 3D mesh with the corresponding 3D coordinates given by the stereo rig, which eventually gives the approximate depth of the object. Other very important aspect of these stereo cameras is the increase of the information of the object from different angle, also known as Field Of View (FOV). Irrespective of the fact that stereo cameras can be used to build approximate 3D model, they can also provide the solution to face alignment by providing multiple views of the object from different angles.

Depth of the object provided by the stereoscopy of the images from a stereo camera is not that accurate. Thus one can exploit the advantage of increased FOV, in the case of multiple or double cameras installed far apart from each other. Figure 1.5 shows the installation of these cameras on the extreme edges of a display screen. This proposition of installation will ultimately help to solve the problem of large lateral movements of a face.

Multiple cameras also suffers from the problem of calibration. Not only intrinsic but extrinsic calibration parameters are calculated after the installation of these cameras. Since this calibration is one time issue therefore the work in this thesis considered that the cameras are well calibrated.

This thesis presents solutions for both single camera and double camera system. Most of the research teams has worked on facial analysis by single camera, therefore

Figure 1.5: Example of double camera installation

results of the proposed algorithms are also compared in a single camera system. Whereas one of the novelty of this thesis is the work done in the multiple camera configuration and how the multiple facial views are handled simultaneously.

1.4 Thesis Organization

A short organization of the thesis is shown in figure 1.6. The blocks highlighted refer to our contributions in the thesis. Description of each chapter is given below.

Chapter 2 starts with the explanation of different deformable model based methods. It then describes the variations implemented by the researchers in these methods. It also discuses the proposed variations in AAM and concentrate on the propositions made to ameliorate its performance in terms of time, memory, efficiency and robustness. Most of these proposed improvements are useful for the problem stated in this thesis i.e. pose estimation and feature extraction of unknown oriented faces.

Chapter 3 explains the basics and preliminary concepts of this thesis. It starts with the construction of a new 2.5D AAM (our first contribution), based on 3D model, which makes it possible to perform pose estimation and features localization of an oriented face. Secondly it gives detailed description of facial image databases both for singe view and multi view camera. Multi view camera setup, calibration and image acquisition are explained in this section. It also explains how the synthetic facial database is acquired both for single and multiple camera systems. Lastly it describes the method for the error evaluations for the experiments.

Chapter 4 presents the solution to the problem stated in this thesis in a single camera configuration using the 2.5D AAM of chapter 3. It presents our second contribution of an efficient optimization technique for AAM by the hybridization of genetic algorithm (GA) with gradient descent (GD) to make a robust, efficient and real time face analysis system.

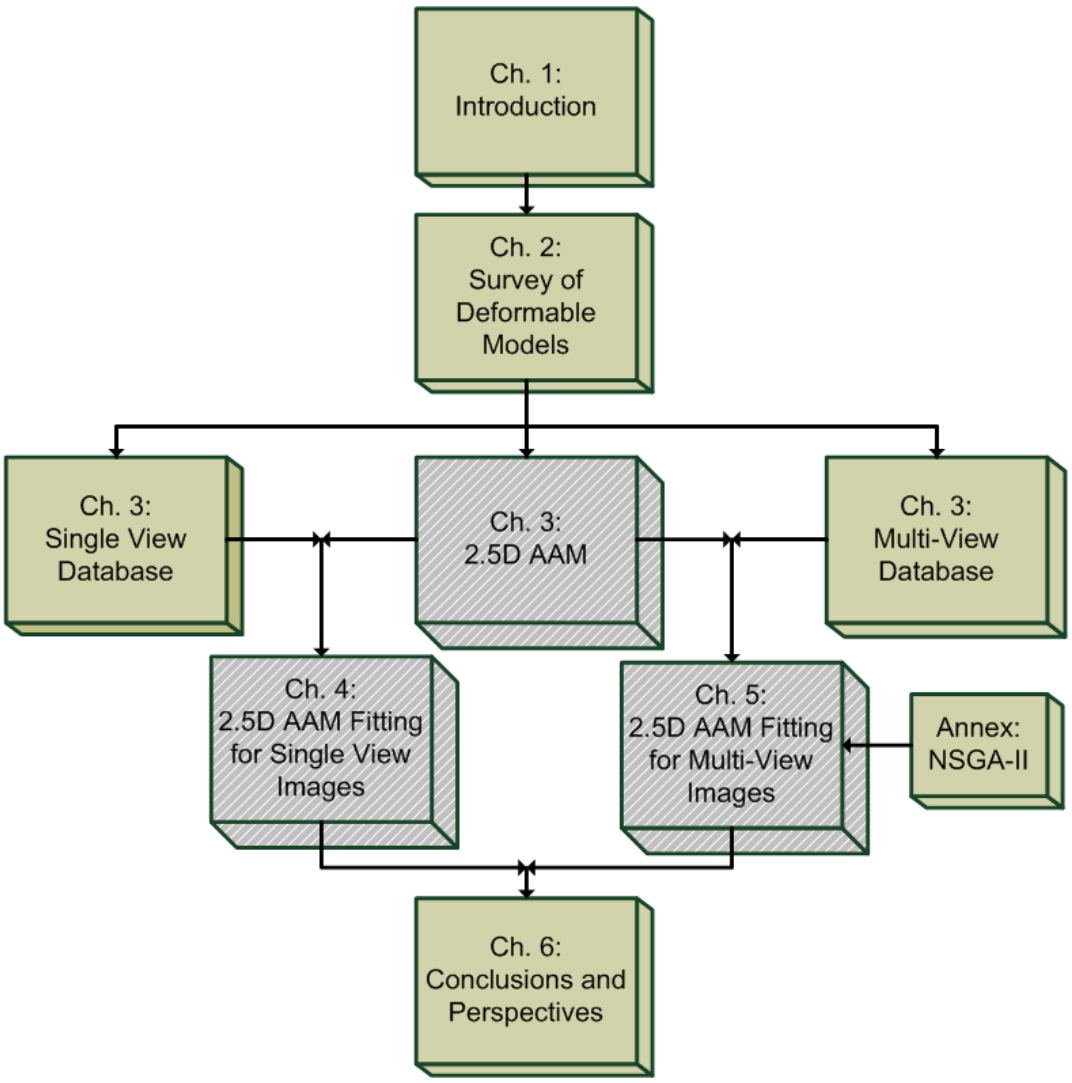


Figure 1.6: Thesis Organization

Chapter 5 presents third contribution of the thesis. It presents the solution to the problem stated in this thesis in multiple camera configuration using the same 2.5D AAM of chapter 3. It describes the analysis of the multi-view facial images by previously proposed 2.5D AAM. It discusses two new concepts of multi-objective and hybrid multi-objective optimization and how it can be implemented in a multi-view face analysis system by 2.5D AAM.

Chapter 6 concludes the work done in the thesis along with the international publications extracted from this work. It also discusses implementation of the algorithms in real life problems. Finally some research perspectives appear at the end of this thesis.

Chapter 2

Survey of Deformable Models

Contents

The deformable model gained a lot of interest in the last decade and researchers have proposed its various versions. This chapter provides a brief introduction to the deformable model based methods, followed by detailed description of classical AAM. This thesis focuses on the AAM based methods, since most of the research teams have used AAM and have improved this method. Section 2.2 describes improvements and expansions made in this domain. Its subsection 2.2.1 discuses the proposed variations in AAM and subsections 2.2.2, 2.2.3 and 2.2.4 concentrate on the propositions made to ameliorate the performance of AAM in terms of time, memory, efficiency and robustness. Most of these proposed improvements are useful for the problem stated in this thesis i.e. pose estimation and feature extraction of unknown oriented faces. Section 2.3 concludes the discussion.

2.1 Deformable Models

Generally deformable model based methods work in two phases. The first phase is the creation of model, which can be used to generate a set of plausible representations in terms of shape and/or texture of the learned objects. The second phase (segmentation phase) is to find the optimal parameters of variation of the model, in order to match the shape and/or the texture of the object in an unknown image. The search and matching of the visual objects in an image requires an optimization of the parameters of the variation of the model. This optimization process is an iterative process, in which model parameters are adapted to minimize the error between image under analysis and the model itself.

2.1.1 Elastic Bunch Graph Matching

Wiskott et al. [22] used a technique called Elastic Bunch Graph Matching where they mapped a deformable grid onto a face image which is comprised of sparsely distributed feature points to make Face Bunch Graphs (FBG) as shown in the 2.1. The nodes of these feature graphs consists of Gabor jets, where each component of a jet is a filter response of a specific Gabor wavelet extracted at a given image point.

In segmentation a function is used to evaluate the graph similarity between an image graph and the FBG. It depends on the jet similarities and the distortion of the image grid relative to the FBG grid. The goal of this function is to find the fiducial points and

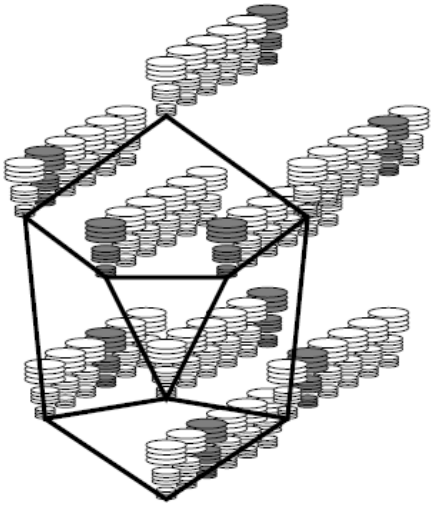


Figure 2.1: Face Bunch Graph

thus to extract from the image a graph which maximizes the similarity with the FBG. In practice, one has to apply a heuristic algorithm to come close to the optimum within a reasonable time. In [22], they used a coarse to fine approach in which the degrees of freedom of the FBG is introduced progressively: translation, scale, aspect ratio, and finally local distortions.

Wiskott et al. [22] used the Gabor Jets of frontal, half-profile and full profile views for face recognition and pose estimation. Maurer and von der Malsburg [27] also demonstrated tracking heads through wide angles by tracking graphs whose nodes are facial features, located with Gabor jets.

The system is effective for tracking, but is not able to synthesize the appearance of the face being tracked. Although, even using reduced images of 128x128 pixels the calculation time for one frame is 4 sec. Moreover system is not able to recover from tracking errors caused by temporary occlusion of features.

2.1.2 Active Shape Models

Active Shape Models (ASM) proposed by Cootes and Taylor [21] constructed by the shapes of all the facial images of a learning database. If a shape is described by n points, it can be represented by n landmark points for a single example as the $2n$ element vector x as

$$x = (x_1, ..., x_n, y_1, ..., y_n)^T$$

All the shapes of the learning database faces are aligned and a mean shape is obtained. A well known data compression technique of PCA is applied on these shapes to obtain shape parameters b as

$$x = \bar{x} + \phi * b$$

where ϕ are the eigenvectors and the vector b defines a set of parameters of a deformable model. By varying the elements of b we can vary the shape x. An active shape model is described by the shape parameters b combined with similarity transformation (pose parameters) defining the rotation θ, translation X_t, Y_t and scale s of the model.

$$x = T_{X_t,Y_t,s,\theta}(\bar{x} + \phi * b)$$

During training phase sampling of the k pixels on either side of the model point is done for every training image. Instead of gray level absolute values, the derivatives of these values are sampled and normalized. During segmentation phase sampling of the m (m>k) pixels on either side of the predicted model point is performed. Then the quality of the fit is tested by comparing them with the gray level model obtained in the training phase. Ultimately the one which gives the best match is chosen.

Recently Milborrow and Nicolls [28] proposed extensions to ASM and used it to locate features in frontal views of upright faces. They extended ASM by increasing the number of landmarks, using patches around landmarks instead of line of pixels and stacking two Active Shape Models in series using the results of the first search as the start shape for the second search.

ASM are robust to illumination variations since they do not involve facial textures at all. On the contrary this becomes one of their drawbacks because textures plays an important role in facial analysis. We feel that significant information is embedded in the texture of face e.g. skin wrinkles, identity etc. Therefore it is necessary to take into account all the texture instead of specific patch or pixels around a facial feature. Moreover in our team, we address the problem of face recognition, face synthesis and face compression for cognitive radio, which requires to include texture information.

2.1.3 3D Morphable Models

Blanz and Vetter [24] introduced a deformable model called 3D Morphable Model (3DMM). Learning database of 3DMM was created by the laser scans of 200 heads of young adults (100 male and 100 female). The laser scans provide head structure data in a cylindrical representation, with radii r(h;ϕ) of surface points sampled at 512 equally-spaced angles ϕ, and at 512 equally spaced vertical steps h. Additionally, the RGB-color values R(h;ϕ), G(h;ϕ),and B(h;ϕ), were recorded in the same spatial resolution and were stored in a texture map with 8 bit per channel. The resultant faces were represented by approximately 70,000 vertices and the same number of color values. The morphable model is based on a data set of these 3D faces. The geometry of a face is represented with a shape-vector $S = (X_1, Y_1, Z_1, ...X_n, Y_n, Z_n)^T$, that contains the X, Y and Z coordinates of its n vertices. The number of valid texture values in the texture map is equal to the number of vertices, therefore the texture of a face is represented by a texture-vector $T = (R_1, G_1, B_1, ..., R_n, G_n, B_n)^T$, that contains the R, G and B color

values of the n corresponding vertices. A morphable face model was then constructed using a data set of m exemplar faces, applying data compression technique of PCA on shape-vector $S_{1..m}$ and texture-vector $T_{1..m}$. Shapes S_i and textures T_i can be expressed as a linear combination of the shapes and textures of the m exemplar faces:

$$S_i = \overline{S} + \sum_{1}^{m-1} \alpha_i s_i$$

$$T_i = \overline{T} + \sum_{1}^{m-1} \beta_i t_i$$

where s_i and t_i are the eigenvectors. α_i and β_i are the coefficients. Apart from these two parameters, there are also rendering parameters ρ which contains camera position (azimuth and elevation), object scale, image plane rotation and translation.

In segmentation phase they use gradient descent algorithm for the estimation of the parameters α, β and ρ for a given input image. The reconstructed 2D image of the model I_{model} is supposed to be closest to the input image I_{input} in terms of Euclidean distance given by

$$E = \sum_{x,y} \|I_{input}(x,y) - I_{model}(x,y)\|$$

where x and y represent the coordinates of the image.

Robustness of these 3DMM is very high compared to other deformable model based method, but its unavoidable drawback is its computational time. According to Blanz and Vetter [29] each face requires more than 4 minutes in a 2GHz Pentium 4 processor. No matter how fast the system is, it requires enormous amount of time to process 70,000 vertices. Therefore they can not be implemented in a real time scenario. Moreover the method of obtaining 3D shapes and textures by laser scanners is cumbersome and expensive due to the requirement of additional hardware.

2.1.4 Candide Model

Candide is a three-dimensional parameterized wireframe model of the human face. It was first created by Rydfalk [26] in 1987, since then it has been updated several times. The last update, by Ahlberg [25], has led to Candide-3 which is shown in figure 2.2. Candide-3 is composed by 113 vertices and 184 triangles.

The configuration of the vertices in Candide-3 is controlled by three different sets of parameters: global, shape and animation. The three dimensional vector containing the coordinates of the model's vertices is denoted by g. The shape and the expression of a face can be expressed by a simple linear equation

$$g = \overline{g} + S\sigma + A\alpha$$

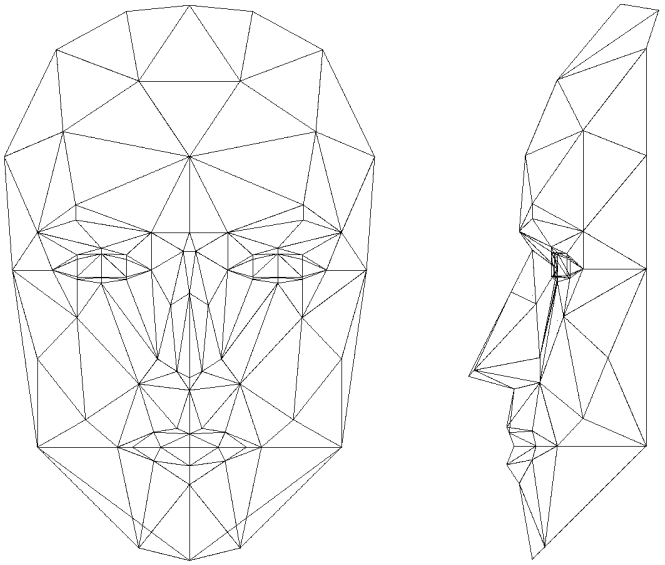


Figure 2.2: Candide-3 Model

where $\overline{g}$ is the vector containing the vertex coordinates of the standard model, while S and A represent the shape and animation units. These are sparse matrices that denote which vertices should be moved (and in which direction) to perform a well-defined deformation of the face's structure. The term $S\sigma$ accounts for the shape variability and σ is the shape parameters vector. $A\alpha$ stands for facial animation and α is the animation parameters vector. Thus by just changing the values in σ the static shape can be controlled (i.e. the distance between the eyes, the mouth vertical position etc.), while α drives the facial expressions (raising the eyebrows, opening the mouth, etc.). To describe the head pose, a global transform is applied to the above formula and therefore the following equation is used instead

$$g(R, s, \sigma, \alpha, t) = Rs(\overline{g} + S\sigma + A\alpha) + t$$

where the global motion is taken into account by $R = (rx, ry, rz)$ (the rotation matrix), s (the scale) and $t = (tx, ty, tz)$ (the translation vector).

To obtain the textures of a facial image, each vertex of Candide-3 model can be manually marked on the face. Since Candide-3 contains 113 vertices and placing them manually on the whole set of training images would be time consuming. A possible approach could be the tunning of the parameters manually to fit the face, it would be much faster than selecting all the vertices manually by hand. For the segmentation phase all the parameters defined above can be evaluated to minimize the pixel difference between this model and the query image. Dornaika and Ahlberg [30] used this face model along with simple gradient descent method as a search algorithm for face tracking.

Candide-3 is, no doubt, a well known deformable face model, but use of large number of vertices (113) makes it inefficient. Moreover the choice and location of the vertices are done by keeping in mind the expression variations. For example, normally the region of the forehead is not visible due to the occlusion by hairs, whereas in Candide-3 this region is given an equal importance compared to other features like eyes, nose and mouth. Therefore this model becomes unsuitable for the solution of the problem stated in this thesis.

2.1.5 Classical AAM

The Active Appearance Models introduced by [31] and [1] in 1998, are the deformable models composed of both shape and texture unlike ASM (Active Shape Model) which are shape deformable model containing only shape. This section will briefly describes the classical AAM for face analysis. The classical Active Appearance Models works in three phases. In the first phase, the model is generated from examples of faces on which points are marked manually and their textures are extracted. All these points and textures are combined and their variations are learned automatically from a principal component analysis. In the second phase (also called as training or pre-computation phase) the model is trained to pre-compute a matrix which helps to find the optimum values of variations with respect to the query images in the segmentation phase. In the third phase it uses its training data for the segmentation of the objects in the query images. Following sections will present the three phases of the AAM algorithm applied on facial image.

2.1.5.1 Modeling

In the first phase, AAM model is generated along with the deformation parameters. A database, called learning database, of facial images is acquired to build AAM. On each facial image of this database, set of points are marked manually. Different researchers have used different number of points on the face. Some of them has included ear while others have surrounded features like nose, eyebrows and ears by bunch of points. The work presented in this thesis have used only 68 points in order to make AAM model time and memory efficient. These 68 point are shown in the figure 2.3. Combination of these 68 points on each face is regarded as a shape. If there are N number of images in the database then the vector representation of these shapes is

$$s_i = [x_{i,1}, x_{i,2}, ...x_{i,68}, y_{i,1}, y_{i,2}, ...y_{i,68}] \ \ (1 \leq i \leq N) \tag{2.1}$$

All the shapes obtained are rotated, resized and translated using Procrustes analysis (Stegmann [3], Goodall [4]). The mean of each point is calculated to create mean shape of 68 points. The mean shape obtained is used to extract and warp the frontal view textures of all the facial images using the Delaunay triangulation as shown in the figure 2.4. These textures undergo the procedure of photometric normalization to normalize their gray levels. Let us suppose the texture g_{image} is the texture of the learning database, then the equation for its normalization would be:

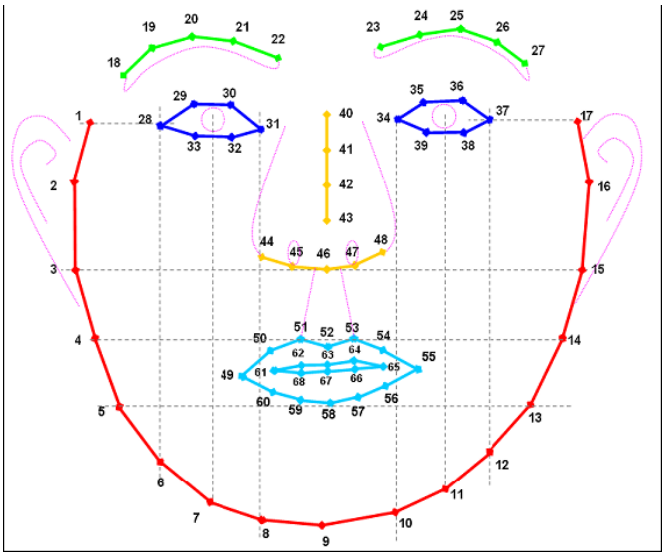


Figure 2.3: 68 landmarks

Figure 2.4: Shape, Texture and Delaunay Triangulation

$$g_{image\_normalisee} = \frac{g_{image} - \beta}{\alpha} \tag{2.2}$$

$$\alpha = \sqrt{\sum_{i=1}^{m} (g_{image}(i) - \overline{g_{image}})^2} \tag{2.3}$$

$$\beta = \overline{g_{image}} \tag{2.4}$$

where α is the standard deviation and β is the mean of the pixels of the texture. This photometric normalization works efficiently on the databases for illumination variations. However, Le Gallou et al. [32] proposed an adaptive histogram equalization technique of CLAHE (Contrast Limited Adaptive Histogram Equalization).

Principal Component Analysis (PCA) compression is applied on the shapes and textures, to obtain shape and texture parameters with 95% of the variation retained. Each shape s_i and the texture g_i of the learning database can be synthesized by these shape and texture parameters with the help of the following equations.

$$s_i = \bar{s} + \phi_s * b_s \tag{2.5}$$

$$g_i = \bar{g} + \phi_g * b_g \tag{2.6}$$

where $\bar{s}$ and $\bar{g}$ are the mean shape and mean texture; ϕ_s and ϕ_g are the shape and texture eigenvectors obtained during PCA; b_s and b_g are the shape and texture parameters respectively.

Both of the above parameters are combined by concatenation of b_s and b_g. And a final PCA is performed to obtain the appearance parameters C.

$$b = [b_s b_g]^T, b = \phi_C * C \tag{2.7}$$

where ϕ_C are the eigenvectors obtained by retaining 95% of the variation and C is the matrix of the appearance parameters, which are used to obtain shape and texture of each face of the database.

AAM model can be translated as well as rotated with the help of pose vector P.

$$P = [\theta, t_x, t_y, Scale]^T \tag{2.8}$$

where θ corresponds to the face rotation and t_x, t_y are the offset values from the supposed origin and $Scale$ is the magnification of the model. Figure 2.5 shows the AAM model deforming and rotating by changing C and P parameters respectively.

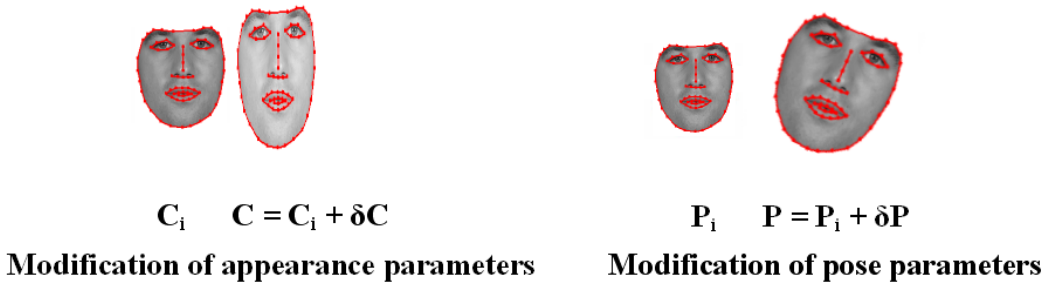


Figure 2.5: AAM by varying parameters

2.1.5.2 AAM Training

The AAM model obtained in the previous section can be used directly for the face search or the search can be directed with the help of the matrix precomputed by training images. Although direct application has some advantages over the trained AAM but for that an efficient optimization technique (e.g. gradient descent, genetic algorithm, Nelder Mead simplex etc.) is required. In the classical method of AAM, model is trained by applying it on the training images while introducing variations in all the parameters one by one. Residual images, which correspond to the difference between the model and the training image, are obtained for each parameter variation.

As explained in the previous section 2.1.5.1, each image of the learning base can be synthesized by a particular value of parameters C and P. Let C_i be the value of appearance parameters of the image i of the learning database and P_i be the value of pose parameters. By changing the parameters C_i and P_i, respectively by δC and δP ($C = C_i + \delta C$ and $P = P_i + \delta P$), a new shape s_m and a new texture g_m (equation 2.7) are synthesized (Figure 2.6).

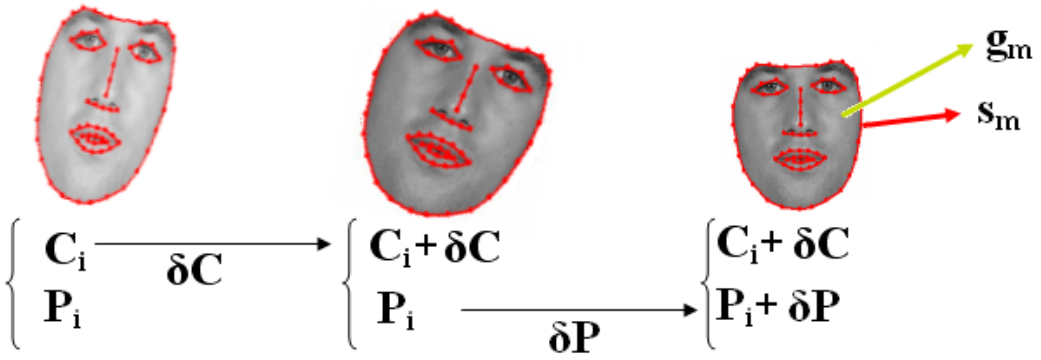


Figure 2.6: Training

Lets consider the texture g_i as the texture of the original image i then pixel difference

or the residual image is given as $\delta g = g_i - g_m$. By varying each parameter at a time and generating its residual images one can create a linear relation between them. This relation is created through principal component regression technique by [33], in order to keep its dimensionality to a feasible size. The relation R_C between δc and δg and relation R_P between δP and δg are given as

$$\delta C = R_C * \delta g \tag{2.9}$$

$$\delta P = R_P * \delta g \tag{2.10}$$

where regression matrices R_C and R_P are of the size $NumberofC \times NbPixels$ and $NumberofP \times NbPixels$ respectively.

In later publication by Cootes et al. [34] and Cootes and Taylor [35] this principal component regression is superseded by a simpler approach. They calculated the partial differential of residual images with respect to each parameter and taking arithmetic mean of these partial differentials for all the training images and k variations in each parameters. This learning approach is denoted as Jacobian and is given as

$$\frac{\partial \delta g}{\partial P_j} = \frac{1}{M} \sum_{r}^{M} \sum_{k} \frac{\delta g_r(P_j + \delta P_{jk}) - \delta g_r(P_j - \delta P_{jk})}{2\delta P_{jk}} \tag{2.11}$$

where M is the number of training images, k is the variation on the j^{th} pose parameters P. Similarly Jacobian are calculated for each C parameters. Now R_P and R_C are calculated as

$$R_P = \left(\frac{\partial \delta g}{\partial P}\right)^{-1} \tag{2.12}$$

$$R_C = \left(\frac{\partial \delta g}{\partial C}\right)^{-1} \tag{2.13}$$

To obtain numerical stability, a singular value decomposition (SVD) of the Jacobian matrices ($\frac{\partial \delta g}{\partial P}$ and $\frac{\partial \delta g}{\partial C}$) are preferred in order to obtain their respective pseudo-inverse R_P and R_C. However due to the size this is not feasible, therefore a normal matrix inversion is carried out.

This approach is no doubt efficient as far as training is concerned especially when the number of training images and/or number of pixels of residual images becomes large enough such that it become impossible to store them in order to apply principal component regression method. Thus this method of training is easier to implement, faster to calculate and requires far less memory to execute. Stegmann [36] observed that this training scheme do not differ significantly, from linear regression, in performance during segmentation. In fact Jacobian are slightly better due to smaller computational and memory demands.

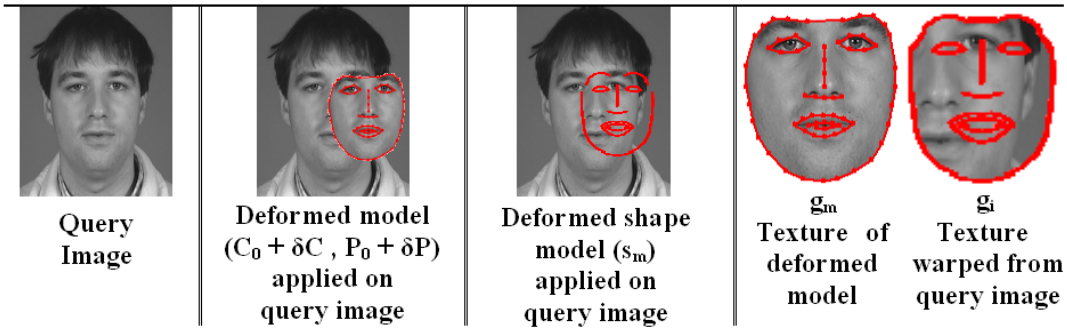


Figure 2.7: Segmentation

2.1.5.3 Segmentation

In segmentation phase the deformed, rotated and translated shape model obtained by varying C and P parameters, is placed on the query image I to warp the face to mean frontal shape. After this shape normalization photometric texture normalization is applied to overcome illumination variations. The objective is to minimize pixel error

$$e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[I_i(C,P) - M_i(C)]^2} \tag{2.14}$$

where $I(C,P)$ is the segmented image and $M(C)$ is the model obtained by C parameters and N is the number of pixels of the model. C and P parameters are calculated by using the relations of equations 2.9 and 2.10.

The search algorithm, illustrated in figure 2.7, is described below.

1. Generate g_m and s from the values of parameters C and P (initially set to 0).
2. Compute g_i, which is obtained by placing the shape s and warping the query image segment to mean frontal shape followed by texture normalization.
3. Calculate the residual image $\delta g_0 = g_i - g_m$, and the residual error $E_0 = |\delta g_0|$.
4. Predict $\delta C_0 = R_C * \delta g_0$ and $\delta P_0 = R_P * \delta g_0$.
5. Find new value of residual error $E_j < E_{j-1}$ with the variations predicted in appearance $C_j = C - k * \delta C_0$ and pose $P_j = P - k * \delta P_0$ parameters. Where k represents the discrete step sizes of 0.25, 0.5, 0.75 and 1.0.
6. Repeat steps from 1 to 5 while $E_{j-1} > E_j$, where E_{j-1} is the residual error of the previous iteration.

When the convergence of the error E_j is reached, i.e. when $E_{j+1} \geq E_j$, parameters C_j and P_j corresponds to the best parameters for the representations of the texture and shape of the face in a query image.

A brief research work on the methods presented in this section was required to search for a method which would be more robust to the deformations of a face, so that facial features and pose of an unknown face could be extracted more efficiently. For this kind of application AAM methods are the most suitable approaches. Although [33] reported that the results of ASM is better than AAM for marker detection, but we feel that significant information is embedded in the texture of face e.g. skin wrinkles, identity etc. Therefore it is is necessary to take into account all the texture instead of specific patch around a facial feature in the case of ASM and EBGM. Moreover in our team, we address the problem of face recognition, face synthesis and face compression for cognitive radio, which requires to include texture information. As far as other methods are concerned, AAM is more rapid than 3DMM and can make the use of 2D images obtained from a camera instead of using laser scanner. Inspite of using predefined Candide model we have created our own AAM model in order to enhance the facial features required for facial analysis. This method has been widely used in various applications of lip-reading, cloning and expression detection etc. This thesis focuses on the method of Active Appearance Models (AAM) and next section presents the state of the art for AAM.

2.2 AAM Advancements

AAM algorithm has proved to be a successful method for matching the model to the query images. Since the classical AAM was described there have been a number of modifications and improvements proposed by several researchers, claiming to be superior than classical AAM. This section and its subsections give a detailed description of these improvements.

Subsection 2.2.1 presents a number of proposed improvements and alternatives to the original classical AAM, including different training methods, non-linear models and different methods of updating the model during the search. The performances of Shape-AAM, Nonlinear AAM, DAM, TC-ASM, TB-AAM and compositional AAM are compared with that of the classical AAM.

Subsequent sections will focus on the methods adapted in AAM, which may be used for the solutions of the problems stated in this thesis i.e. pose estimation, features extraction of an unknown oriented face. These approaches can be divided into the three following subsections. Subsection 2.2.2 presents the work done, for the facial analysis of the oriented face, by the extension of the AAM model and its appearance parameters. Subsequent subsection 2.2.3 describes the facial analysis by fitting AAM on temporally synchronized multiple images acquired from two, three or multiple cameras. Subsection 2.2.4 discusses different optimizer used by the research teams for the face search optimization in the segmentation phase of AAM.

2.2.1 AAM Variants

The classical optimization of AAM by linear regression presents some drawbacks. It needs to store the matrices R_C and R_P in memory for the segmentation phase. Moreover the precomputations from the training set is only an approximation for any given target image, and may be a poor one if the target image is significantly different (unknown or unseen faces) from the training images, as discussed and tackled by Cootes and Taylor [37]. A number of improvements and alternatives to the original classical AAM proposed by the research teams are discussed in this section.

2.2.1.1 Shape-AAM

The Shape-AAM has been proposed by Cootes et al. [38]. It's an alternative approach to use residual image to compute the shape and pose parameters, while the texture parameters are computed directly from the image with the help of the shape model. The statistical model is created on the shape and texture, without the concatenation of two parameters (3rd PCA). Instead of appearance regression matrix of R_C, shape regression matrix R_s is calculated along with pose regression matrix R_P.

This method is more time-consuming (Cootes and Kittipanya-ngam [39]) than the classical method of AAM, and may be useful when there are few shape modes and many texture modes.

2.2.1.2 DAM

The direct appearance model, introduced by Hou et al. [40], also removes the 3rd PCA in classical AAM modelization i.e. they do not combine shape and texture parameters of AAM. Unlike the Shape-AAM of Cootes et al. [38], they use information of the texture instead of the shape. They considered that the texture and shape are sufficiently correlated and the shape can be obtained from the texture of the model through a relationship built during the learning phase of DAM:

$$b_x = Sb_g \tag{2.15}$$

Therefore texture regression matrix R_g and pose regression matrix R_P are calculated in the training phase. The segmentation procedure is similar to classical AAM except they evaluate shape from the texture by the equation 2.15.

Although it gives better results than the classical AAM method of [39], but it requires prior information of the correlation of texture and shape. Therefore this texture and shape linearity makes it difficult to analyze unknown oriented faces. For example if there are large number of different faces in the learning database, we are not confident in the equation 2.15, since it is possible that two totally different textures (with or without beard) might produce the same shape.

2.2.1.3 Nonlinear AAM

Romdhani et al. [41] have extended AAM to nonlinear cases across very large pose views based on Kernel Principal Component Analysis (KPCA). These nonlinear models represent the corresponding dynamic appearances of both shape and texture across pose. In other words, a non-linear 2D shape model is combined with a non-linear texture model on a 3D texture template. The approach is promising, but the computation involved is rather intensive.

Donner et al. [42] replaced PCA of multivariate regression in training phase of classical AAM by Canonical Correlation Analysis (CCA). The method learns a linear regression between the canonical projections of the residual images and parameter variation. The method utilizes canonical correlation analysis to find the subspaces which best adheres to a linear regression. They have shown that their approach eliminates the need for using different step sizes (see step 5 of section 2.1.5.3), as in the case of classical AAM, and claimed that their approach is four times faster than classical AAM as the parameter predictions are more accurate. The results of this technique is similar to the classical AAM.

2.2.1.4 TC-ASM

Yan et al. [43] proposed a novel morphable model technique called Texture Constrained Active Shape Model (TC-ASM). It inherit the merits and reject the demerits of ASM and AAM. They borrow local appearance models from ASM for the landmark localization and incorporate the global texture constraint over the shape from AAM for more accurate shape parameters estimation. In each iteration of face search, a better shape is found under Bayesian framework.

This method is an extension of DAM (see section 2.2.1.2). Although it is claimed in [39], to provide more accurate results than ASM and classical AAM. However it works in those cases in which shapes are correlated with the textures, particularly in medical images analysis problems. Whereas in face analysis, since different textures can produce same shapes, therefore recognition and synthesis of a face could not be accomplished with this method.

2.2.1.5 TB-AAM

Recently Lee and Kim [44] gave a new concept of Tensor-Based AAM. This concept is based on the *tensor* which is also known as n-way array or multidimensional matrix or n-mode matrix. It is a higher order generalization of a vector (first order tensor) and a matrix (second order tensor). They incorporated a series of learning databases images with different identities, expression, pose and illumination variations etc. To include these variation by specific basis vectors they make the use of multi-linear algebra of Alex et al. [45] for the multi-linear analysis of the images with variations defined above.

Alex et al. [45] introduced a counterpart of "'eigenfaces"' as tensorfaces which combines several modes of variations in facial images. Alex et al. [46] performed facial recognition with these tensorfaces and compared it with eigenfaces based approach. Lee and Kim [44] used these tensorfaces concept in AAM modeling. Thus they are able to generate variation specific AAM model. In which pose and expression variation is incorporated in shape model and illumination variation in appearance/texture model. For the fitting phase they estimate the pose, expression and illumination condition before hand and construct the respective AAM model for fast fitting. Although their fitting process is similar to conventional AAM but their model enables them to converge more rapidly and efficiently.

The main drawback of this technique is the computations to select the model for the current query image, therefore takes more time despite of the fact that it converges more rapidly and efficiently.

2.2.1.6 Compositional Approach for AAM Fitting

Matthews and Baker [47] and then Mercier et al. [48] used compositional approach for active appearance model. Their fitting algorithm is based on gradient descent algorithm called Inverse Compositional Lucas-Kanade algorithm (IC-LK) proposed by Baker and Matthews [49]. Their model generation procedure is similar to AAM but with out performing the concatenation of the two parameter i.e. 3rd PCA. Equation for the AAM after performing the PCA is obtained as

$$s_i = s_{mean} + \sum_{k=1}^{NbS} \Phi_{s_k} * b_{s_k} \tag{2.16}$$

$$g_i = g_{mean} + \sum_{k=1}^{NbG} \Phi_{g_k} * b_{g_k} \tag{2.17}$$

In the segmentation phase, the texture of the query image I inside the shape of the current model s is extracted and warped. The residual image is calculated as $\delta g = g_i - g_m$, where g_m is the texture of the current Model. Followed by the calculation of the steepest descent images $SD = \nabla g_m \frac{\delta W}{\delta b_s}$ with respect to the variation of each shape parameter b_s. Then the Hessian Matrix H is calculated as: $H = \sum_{pixels} SD^T SD$.

In segmentation phase the parameters b_s and b_g are updated by the equations $[\Delta b_s, \Delta b_g]^T = -H^{-1} \sum SD^T \delta g$ iteratively until the convergence is achieved.

The results obtained by this approach are equivalent to the classical method of AAM by linear regression, however, it introduces the complex calculations of the gradient descent and the Hessian. Cootes and Kittipanya-ngam [39] has shown that this approach is more time consuming than the classical method.

2.2.1.7 Active Wavelet Networks

Hu et al. [50] proposed a method for face alignment called active wavelet networks (AWN), which replaces the AAM texture model by a wavelet network representation. They proposed that since PCA-based texture model of AAM causes the reconstruction error to be globally spread over the image and their model consider spatially localized wavelets for modeling texture, therefore the alignment of the face by AWN is more robust to occlusions and variations in illumination. This methods is not suitable for classical AAM but can be adapted for the method of Shape-AAM.

In each iteration they need to calculate texture parameters by orthogonally projecting the normalized face image into the learned wavelet subspace of the training phase. Texture reconstructed from these textures parameters is used to calculate residual image. Thus the use of Gabor wavelet filters make this algorithm very complex.

2.2.2 Model Extension

This section presents the first approach to estimate the facial pose and features by the extensions of the AAM model and its appearance parameters. One way is to use multiple 2D AAM model each corresponding to different face orientation (section 2.2.2.1). Other way is to use a single 3D AAM model which can be rotated with the help of the pose parameters to compensate each facial orientation (section 2.2.2.2). Some researchers have also extended appearance parameters for the pose variability (section 2.2.2.3).

2.2.2.1 Multiple 2DAAM Model

One way of estimating facial pose and features is by generating multiple AAM models in such a way that each corresponds to the specific orientation of the head. In segmentation the matter is to select the required AAM model which could be matched with the facial pose of the query image.

Cootes et al. [51] showed that by using five models it is sufficient to deal with faces that range varies 180 degrees (from left profile to right profile). To adapt to the pose of a face, five models are built from different learning databases: a learning database of left profile faces ($-90°$), a learning database of left semi profile faces ($-45°$), a learning database of frontal view faces ($0°$), a learning database of right semi profile faces ($45°$) and a learning database of right profile faces ($90°$). Thus in segmentation, for each query image, five models are then used and only the model providing the lowest error convergence is considered.

Similarly Cootes and Taylor [33] and Shan et al. [52] performed pose prediction by using three AAM models, one dedicated to the frontal view and two for the profile views. Sung and Kim [53] detected pose-robust facial expression by using three 2D+3D AAM models, one dedicated to the frontal view and two for the side views. Li et al. [54] also used three DAMs (Direct Appearance Models) for face alignment.

Obviously the accuracy of the pose estimation keeps on increasing as the number of AAM model increases. Xin and Ai [55], Liu et al. [56] implemented Active Shape Model (ASM) for the face alignment, by using five poses of each face to create a model. Feng et al. [57] have specified seven models corresponding to seven different facial expressions. Romero and Bobick [58] used another appearance based architecture employing five view-specific template detectors to track large range head yaw by a monocular camera. The Radial Basis Function Network interpolates the response vectors obtained from normalized correlation from the input image and five template detectors.

Peyras et al. [59] also used pools of AAMs, each AAM is being specialized on a particular pose and expression. They separated the sources of variability within the learning database, by dividing it into smaller databases, such that a single database contains images from all the identities with same pose and expression. In this manner all the smaller databases have their respective constant poses and expressions. Therefore each AAM of the pool generated from each learning database is specialized to particular pose and expression. For the segmentation phase they used optimization framework of inverse composition algorithm of Matthews and Baker [47].

Yang and Byun [60] proposed a method of multiple AAM each corresponds to a part of the training database. A fixed mean precomputed Jacobian matrix is not a good choice when the distribution of a training database is nonlinear because the mean can not represent the variation of a training database. They proposed multi-subspaces AAM in which they divide a training database into multi-subspaces along the illumination direction, and build the independent AAM for each subspace. At a fitting phase, they adaptively choose a subspace well fit to a target image.

Finally, Hu et al. [61] have also applied this method for active wavelet networks (AWN), a deformable model similar to the AAM in which changes in texture of the learning database are modeled with wavelet networks.

Use of more than one model of AAM has some disadvantages: i) Storage of shapes and textures of the images of all the models requires an enormous amount of storage memory. ii) Extensive processing of computing several AAM in parallel to determine the model required for query images, eventually makes the system sluggish. Moreover classical AAM search methodology requires precomputed regression matrices, which become a burden on time and memory as the amount of training images increases.

2.2.2.2 Multi-Dimension AAM Model

To introduce the pose variability in AAM, 3D AAM methods are proposed to model the face in three dimensions. This allows to learn and build a model with parameters controlling the variations in facial pose. In these methods a set of images are annotated in three dimensions to modelize a face. Various types of three dimensional model exists such as 3D AAM (obtained from 3D scanner), 2D+3D AAM (a combination of 2D and

3D AAM) and 3DAMB AAM (a combination of muscle-based and anthropometry-based face model). The following subsections explain these multidimensional AAM in detail.

3D AAM Dornaika and Ahlberg [30] used 3D face model Candide along with simple gradient descent method as a search algorithm for face tracking. They used a Candide shape model (as explained in section 2.1.4) to extract the textures of the learning database images in order to build 3D AAM model. Although the Candide shape model is able to introduce both shape and expression variations but these variations are synthetic and not related to real facial images. 3D facial tracking performed in [30], requires a preprocessing step of extracting the texture by mapping Candide shape model on the frontal view of the user. Since this texture is used through out the tracking, therefore the system is unable to track any unknown face.

Sung and Kim [62] applied 3D AAM for face tracking in a video sequence using IC-LK algorithm of Matthews and Baker [47]. They created the 3D AAM model by annotating the face in a movie created from a stereo vision camera and implemented a stereo vision technique to relate the landmarks on different images. Originally IC-LK algorithm was developed for the face search by 2D+3D AAM. Therefore, for their 3D AAM, they modified the IC-LK algorithm by redefining warping function, inverse of warping and composition of warping function. Facial tracking by this method is again person dependent and unable to work for unknown faces.

Von Duhn et al. [63] used three cameras to acquire three (frontal, profile and angle) views of a face. Landmarks localized by 2D AAM on each image are correlated to build a 3D model. They used this model for the recognition of an oriented face.

Paterson and Fitzgibbon [64], Blanz and Vetter [29], Ishiyama et al. [65], Malassiotis and Strintzis [66] also developed 3D models of faces, made from faces acquired by laser scanners providing cylindrical data of the face. In order to create the 3D deformable model, a morphing is performed between all these examples of 3D faces. The deformations of the model are controlled by parameters such as of AAM. This method requires a learning 3D faces of good resolution.

2D+3D AAM Xiao et al. [67], Hu et al. [68], Koterba et al. [69], Ramnath et al. [70] used 2D+3D AAM along with a fitting algorithm, called inverse compositional image alignment algorithm, which is an extension of a gradient descent method. Their AAM model is obtained by the non-rigid structure-from-motion algorithm of Xiao et al. [71]. This algorithm requires 2D shapes by tracking the face in a video sequence by 2D AAM, followed by the computation of 3D shape modes from this 2D AAM shape. Ultimately they combined these 3D shape modes with 2D AAM to build 2D+3D AAM model. Their fitting algorithm is similar to Matthews and Baker [47] with additional 3D shape mode to optimize. Sung and Kim [53] also used 2D+3D AAM of Xiao et al. [67] to detect pose-robust facial expression by using three 2D+3D AAM models, one dedicated to the frontal view and two for the side views.

Those techniques of building the 3D model requires structure-from-motion algorithms, by applying an efficient 2D AAM on the sequence of oriented facial images. This procedure do not provide enough accurate 3D model compared to manually labeling the landmarks on the frontal and profile views of a facial image. Additionally, number of shape parameters for a face search optimization are increased, compared to a simple 3D AAM with increased pose parameters.

3DAMB AAM Cordea and Petriu [72] combine muscle-based face model and anthropometry based face model to create 3D Anthropometric-Muscle Based AAM (3DAMB AAM). The 3D model uses muscle actuators to model facial expressions and anthropometrical controls to model facial types. The shape model variations, caused by these controls, are used to extract the textures of the individuals in the database. Followed by creating shape, texture and appearance parameters, similar to the classical AAM. The segmentation phase is also similar to the classical AAM.

In Martins and Batista [73] they used AAM combined with Pose Orthography and Scaling with ITeration (POSIT) for the pose estimation. They localize facial features by classical AAM and used statistical anthropometric 3D model for the evaluation of the pose by POSIT. As classical AAM is unable to estimate large lateral movements of a face, therefore this method is also limited to estimate with in -15 to +15 degrees of profile angle variation. For the estimation of small pose variations, 3D AAM is better than using POSIT, as both (POSIT and additional pose parameters of 3D AAM) of them required additional processing time.

In these methods, the use of anthropometric 3D model makes the system complex. In addition, a 3D AAM made from real faces is more robust in terms of feature and expression detection (due to their natural feature and expression variations) compared to this anthropometric 3D model.

2.2.2.3 Appearance Parameter Extension

One way of estimating the pose is by considering the facial pose as appearance variations. Coupled View AAM is used in Cootes et al. [74] to estimate the pose profile angle. In the training phase they include 2D shapes and 2D textures of both frontal and profile views of each subject. Appearance parameters of their CV-AAM have the capability to estimate the profile angle. Appearance parameters of their model can tune both the shape and the profile angle of a face. In other words facial pose angle variation is considered to be facial appearance variation, eventually evaluating the pose by tuning the appearance parameters.

For the profile angle estimation they have combined both frontal and profile view. The texture and shape vectors are twice larger than in classical AAM. Instead of combining appearance parameters of two views and optimizing several parameters, it is far better to use 3D AAM with one pose parameter for profile angle estimation.

2.2.3 Multi-View Images

This section presents the second approach to estimate facial pose angles and features by fitting the above 2D, 2D+3D or 3D deformable models on multiple images acquired by two, three or multiple cameras. In single-view system face alignment cannot be accomplished when a face occlude itself during its lateral motion. Such as in a profile view only half of the face is visible. To overcome this dilemma a multi-camera approach can be adapted.

Facial images from multiple cameras can be used in two ways. One possibility is to fit AAM on these images simultaneously and acquire the facial orientation by using the weak-perspective camera model (section 2.2.3.1). Secondly, AAM fitting is accomplished on the disparity data obtained from stereos cameras to estimate the facial orientation (section 2.2.3.2).

2.2.3.1 Simultaneous AAM Fitting

In single-view AAM, single error between model and query image is optimized. However in multi-view AAM, optimization of more than one error is to be performed between a model and query images from each camera.

Hu et al. [68] proposed MVAAM (Multi-View AAM) a robust algorithm of fitting a 2D+3D AAM to multiple images acquired at the same instance. Koterba et al. [69], Ramnath et al. [70] extended the work of Hu et al. [68] by incorporating camera calibration in MVAAM. In the first part cameras are calibrated using MVAAM fitting on human faces instead of calibration grids. In the second part, improved performances of MVAAM are calculated and compared with the uncalibrated multi-view fitting. Kim and Sung [75], Sung et al. [76], Sung and Kim [77, 78] proposed another algorithm of face tracking by Stereo Active Appearance Model (STAAM) fitting, which is an extension of the fitting of 2D+3D AAM to multiple images.

Their fitting methodology, instead of decomposing into multiple independent optimizations from multiple cameras, adds all the errors. Thus this addition of errors loses the importance of usage of multiple cameras. Moreover they used compositional approach for AAM fitting, which eventually requires complex pre-computations of Jacobian and Hessian matrix.

Romeiro and Zickler [79] used 3D morphable model (as explained in 2.1.3) for the recovery of face from stereo pairs of images in the presence of foreign body occlusion. Model-fitting is performed by finding pose, shape, texture and illumination parameters simultaneously on the two images from each camera. For the optimization they have used quasi-Newton gradient descent method.

These techniques added the errors resulted from analysis of facial images from multiple cameras. This addition is might be due to the incapability of their face search optimization methods to handle multiple errors from multiple cameras. Addition of these errors could cause the system to fail, if with respect to one of the camera, face is

oriented such a way that it does not deliver valid information. High error values from this camera could cause deterioration of the results even if other cameras provide valid information. Therefore, for better results, it is necessary to discard information from such cameras.

2.2.3.2 AAM Fitting by Binocular Disparity

In Dornaika and Sappa [80] the advantages of adaptive appearance model based method is combined with a 3D data-based tracker using sparse stereo data. They used their 3D AAM for the rough estimation of the 3D pose of the head in a video stream. Followed by the improvement of head pose by the sparse stereo 3D data from a stereo rig. The mesh obtained by the appearance based tracker undergoes 3D registration with the corresponding 3D coordinates given by the stereo rig. For 3D registration they used RANSAC-like technique.

Liebelt et al. [81] performed the similar approach by combining the AAM fitting on 2D images and 3D shape alignment on disparity data obtained from stereo cameras. They used 2D+3D AAM model of Xiao et al. [67].

Mittrapiyanuruk et al. [82] also applied AAM for the pose estimation of rigid objects by stereo cameras. They applied 2D AAM on both the images from each camera and used a simple linear 3D reconstruction method of Faugeras [83]. Followed by 3D registration of 3D scene points in the camera coordinate frame with the 3D model obtained in the training phase of AAM to evaluate pose.

Yang and Zhang [84] proposed a model-based stereo head tracking algorithm and is able to track six degrees of freedom of head motions. They track features from each camera and use epipolar geometry to create a 3D model for the evaluation of the pose. Their face model contains 300 triangles compare to 113 triangles usually used in classical AAM and ICLK based AAM etc. Moreover their initialization process requires user intervention.

Tu et al. [85] performed 2D head tracking for each subject from multiple cameras and obtained 3D head coordinates by triangulation. Sung and Kim [62] and Von Duhn et al. [63] also used images from multiple cameras to build 3D AAM model by correlating the landmarks of each image. Slight calibration error massively deteriorates the triangulation. Furthermore, lack of ground truth error calculations creates uncertainty in the accuracy of their system.

In these techniques, stereo vision based methods (epipolar geometry and triangulation) are used to register 3D mesh with the corresponding 3D coordinates given by the stereo rig, which eventually gives the approximate depth of the object. Moreover these techniques are highly sensitive to the calibration of the cameras. Slight calibration error could result in highly deformed implausible faces.

2.2.4 Optimization

This section presents the third approach to estimate facial pose angles and features by using different optimizers for the optimization of face search in the segmentation phase of AAM. Facial search space formed by the AAM pose and appearance parameters is highly complex and scattered. Especially in 3D AAM these pose parameters represent six degrees of freedom (6DOF) of a face instead of 4DOF in 2D AAM. As long as these pose parameters are restricted within the specified values the error curve between the model and the query image remains convex. System will consider the query image as a face image due to a convex error curve. But when the pose parameters increases its span, e.g. profile view of face, error curve will have several local minima. Various local minima in error function makes gradient based methods inefficient and eventually loses its robustness. Initializations around every local minimum can lead to better convergence by these methods. But the amount of required initializations are so huge that it is impractical to use these methods. However instead of these initializations, it is far better to use direct search methods which exploits and explores the error curve without falling into these local minima.

2.2.4.1 AAM Fitting by Direct Search Methods

The main difference between gradient based search and direct search is their capability of error function exploration. Gradient based methods always tends towards the better solutions while exploiting the current solution and converging towards the gradient of the function. Whereas direct search methods, unlike gradient based methods, also explore other solutions of the function. Their ability to anticipate, that inaccurate or unwanted solutions can lead to the global minimum, differentiate them from gradient based methods. In a multidimensional and a multi-parameterized functions, they initially launch multiple solutions in all dimensions to analyze the error function. In the subsequent iterations they not only search for best solution but also keep records of inaccurate solutions for next iterations. Genetic algorithm and simplex are some of the known direct search methods.

Simplex Nelder Mead Simplex algorithm by Nelder and Mead [86], is an iterative direct search method and it is used to optimize both the appearance and pose parameters at the same time. In these methods AAM model is constructed in the same manner as explained in section 2.1.5.1. For the segmentation phase, $numberofparameters + 1$ solutions are initialized by choosing parameters in each solution randomly and their respective pixel errors are calculated. In each iteration, parameters in each solution are modified i.e. reflected, expanded, contracted or shrunk based on the previous value of the pixel error. It converges when no further good solutions are possible. Fixed number of iterations can be chosen as the stopping criteria. This number is either fixed by number of convergences or processing time.

Our research team in [87, 88] have used Nelder Mead Simplex for the optimization in

2D AAM. Similarly Paterson and Fitzgibbon [64] also used Simplex as an optimization for model-based head tracking technique. Cristinacce and Cootes [89] also used simplex for the optimization of their Template Selection Tracker (TST) to localize the facial features.

Genetic Algorithm Genetic Algorithm is a well known direct search method given by Goldberg [90]. In segmentation phase of AAM appearance C and pose parameters P are considered as genes. All the genes of C and P are concatenated to form a chromosome. Population of these chromosomes is randomly created. Tournament selection is applied to select parents chromosomes from the population to undergo reproduction. Two points crossover and Gaussian mutation is implemented to reproduce next generation of the chromosomes. Thus new generation of the same size of population is created using genes of the fittest of the old chromosome. Elitism can also be implemented to preserve the best possible solution at all time. After calculating a number of generations the algorithm can be stopped according to the specified stopping criteria.

McIntosh and Hamarneh [91] and Ghosh and Mitchell [92] performed segmentation of medical images using genetic algorithm. Stegmann [3] also used an optimization technique inspired from genetic algorithm, but claimed not using mutation and crossover operators. Therefore it can be viewed as a random search technique. Hill et al. [93] used GA as a global search methodology to extract the biological structures in medical images. They combined GA global search with local search of ASM to improve the convergence speed of the search methodology. Local search of ASM refers to the same procedure of linear regression method of classical AAM, however they used only shape model of ASM.

These optimization methods are slower than the classical AAM, but they reduce the required memory space because they do not need to save the regression matrix R_C and R_P in memory. They also improve the efficiency of AAM since exploration of the search space is not restricted as in the case of gradient descent and linear regression methods. Since they do not need training or pre-computation phase, therefore one of the major advantages achieved is the generality. Generality is the capability of the model to analyse other than those faces from which it has been created.

2.3 Conclusions

In this thesis the work has been proposed to solve the problem of face pose estimation and facial features localization of unknown and oriented faces in a Cognitive Radio equipment, as the user is not obliged to remain in frontal view in front of the camera. Similarly facial features localization is also required by CR equipment in order to point out the important regions for the compression of facial information.

Taking into account these constraints, 3D AAM (section 2.2.2.2) is far better than using any other extension of 2D AAM model (section 2.2.2) for the pose estimation of

a face's out-of-plane rotation. The inclusion of extra pose parameters in the case of 3D AAM to compensate the orientation of the face, makes 3D AAM reliable compared to 2D AAM, which try to estimate the pose either by using extra appearance parameters or using various oriented 2D AAM models. As far as AAM fitting on multiple views are concerned, some research teams have used multi-view AAM (section 2.2.3), but their fitting methodology, instead of decomposing into multiple independent optimizations from multiple cameras, adds all the errors. This addition of errors loses the importance of usage of multiple cameras. However capturing facial images from multiple cameras leads to multiple error functions, therefore searching for an optimum solution of a single task employing two or more distinct error functions requires multi-objective optimization (MOO). When faces make large lateral movements the error curve between the model and the query image does not remain convex. This non-convex error curve makes deterministic methods used by various researchers unreliable. Thus face search optimization of classical AAM (section 2.1.5) and gradient based methods (2.2.1.6) does not remain a better choice for the face analysis problem. On the other hand direct search methods discussed in section 2.2.4.1 can efficiently cater the non-convexity of the error curve.

To estimate the facial pose and extract the facial features, our first contribution is a new method of building a 3D AAM model, called as 2.5D AAM (chapter 3) is proposed in this thesis in order to make modelization time and memory efficient. While choosing among single camera or multiple camera, no doubt multiple camera system can acquire facial images with large lateral facial movements due to its increased field of view. But the use of multiple cameras is not common in real life. Therefore this thesis tackle the said problem for both the cases of single and double camera configuration. For the face search optimization problem in single camera configuration, a hybrid optimization of direct search and gradient based methods is proposed as our second contribution (chapter 4). Whereas in multiple camera configuration hybrid multi-objective optimization is proposed (as our third contribution) for the search method by 2.5D AAM to deal with multiple facial informations separately (chapter 5). A comparison between the existing methods and our propositions is tabulated in table 2.1.

Methods	**Face Orientation**	**Unknown Faces**	**Facial Features**	**Time**
Classical AAM	+	+	+	+++
AAM Variants	+	+	+	+++
3D AAM	++	+	+	++
Multiple 2D AAM	+++	+	++	++
HGOAAM[1] (Chapter 4)	+++	+++	++	++
HMOAAM[2] (Chapter 5)	++++	+++	++	+

Table 2.1: Comparison of methods with respect to the problem stated in this thesis. [1] Hybrid Genetic Optimization for 2.5D AAM. [2] Hybrid Multi-objective Optimization for 2.5D AAM.

Next chapter will present the detailed description of the modelization of 2.5D AAM and will discuss different facial image databases. It will also present and explain the setup of multiple camera system to build indigenous multi-view database.

Chapter 3

2.5D AAM and Facial Image Databases

Contents

This chapter provides a detailed explanation of the first contribution of this thesis i.e. the generation of 2.5D AAM, which have been used through out the experiments for the pose estimation and feature extraction of the faces making large lateral movements. This chapter also focuses on the databases used for the facial analysis by 2.5D AAM. Two types of test databases are discussed; mono-view facial images database and multi-view facial images database. Discussion on mono-view images databases is brief due to their vast availability. On the contrary for multi-view images databases, it gives detailed description along with the multiple cameras setup to acquire temporally synchronized real and synthesized facial images from two cameras.

3.1 2.5D AAM

This model belongs to the family of multi-dimensional AAM models discussed in section 2.2.2.2 of chapter 2, which includes 3D AAM, 2D+3D AAM, 3D anthropometry-based face model and 3D Candide face model. We call our model as 2.5D AAM because a real 3D model is the one used in medical image analysis, which also contains cross sectional details along with the surface. It can be obtained by a typical 3D data set, grouped by 2D slice images acquired by a CT (Computed tomography) or MRI (Magnetic Resonance Imaging) scanner. Instead of pixels, voxels are used in its real 3D mesh.

There are other multi-dimensional AAM models proposed by the researchers. Dornaika and Ahlberg [30] used shape of the 3D face model Candide, to extract textures from the images in order to make 3D AAM model. Sung and Kim [62] also created the 3D AAM model by annotating the face in a movie created from a stereo vision camera and implemented a stereo vision technique to relate the landmarks on different images. Von Duhn et al. [63] used three cameras to acquire three (frontal, profile and angle) views of a face. Landmarks localized by 2D AAM on each image are correlated to build a 3D model. However, the procedure involved in building 2.5D AAM is simple and fast. It requires only frontal and profile views of an individual compared to sequence of images either by single camera or stereo camera. Manual markings of the landmarks, usually used by research teams, makes it more precise rather than using a semiautomatic procedure by fitting 2D AAM whose robustness may come in question. Section 2.1.5.1 of previous chapter elaborated the steps to generate a 2D AAM model used by classical AAM. While this section concentrates on the first contribution of constructing 2.5D AAM and discusses its construction step by step.

3.1.1 3D Landmarks

In a 2.5D AAM both profile and frontal views of the person is used to make a 3D shape as shown in figure 3.1. 68 points are marked manually on the frontal view of a facial image. In a profile view of the same facial image, only 39 points are visible among the 68 frontal view points. These 39 points are also marked manually on the profile view of a facial image. The formation of these 68 points is well known as most of the researchers [1, 30, 47] have used almost the same kind of arrangement. The optimal distribution of

the frontal view 68 points are configured taking into account the extent of deformation level of each feature as shown in the figure 2.3.

- 17 points surrounds the face to incorporate movements of jaws.
- Two sets of 5 points for each eyebrow to manage their movements in different expressions.
- Two sets of 6 points for each eye to cope with their blinking.
- Only 4 points for nose as it is the least deformed feature in a face.
- 5 points represents the area under the nose because it usually deforms in certain expressions.
- Two sets of 12 and 8 points corresponding to outer and inner boundaries of lips respectively. Since lips are the most highly deformable feature in a face, therefore they are dealt with 20 points.

As shown in the figure 3.1, XY coordinates of all the points of the frontal view are acquired and point corresponding to the nose-tip is taken as the center of these points. Whereas in profile view, X coordinates of all the visible points are aligned with respect to this nose-tip coordinate to obtain the depth of the model. Later on this depth is referred to as the Z coordinates of all the points to build a 3D shape model. Center of gravity (COG) of all these points is calculated, which functions as a pivot for this model. Let us suppose there are N number of images in the database then the vector representation of these shapes is

$$s_i = [x_{i,1}, x_{i,2}, ...x_{i,68}, y_{i,1}, y_{i,2}, ...y_{i,68}, z_{i,1}, z_{i,2}, ...z_{i,68}] \ (1 \leq i \leq N) \tag{3.1}$$

3.1.2 Shape Model and Parameters

All the landmarks obtained in the previous step are resized and aligned in three dimension using Procrustes analysis proposed by Goodall [4]. Rest of the model creation is similar to the one proposed by [1]. Mean of these 3D landmarks is calculated which is called mean shape. Principal Component Analysis (PCA) is performed on these shapes to acquire shape parameters with 95% of the variation stored in them.

$$s_i = \bar{s} + \phi_s * b_s \tag{3.2}$$

where s_i is the synthesized shape, $\bar{s}$ is mean shape, ϕ_s is a matrix whose column represents the eigen vectors obtained during PCA and b_s is a vector of shape parameters. Shapes synthesized by incorporating first three shape parameters are shown in figure 3.2. Each column shows shapes synthesized by varying respective shape parameter from $-3\sqrt{\lambda_i}$ (left), mean shape (center) and $+3\sqrt{\lambda_i}$ (right), where λ are the eigen values corresponding to each shape parameter (obtained during PCA) and i is the index of the shape parameter.

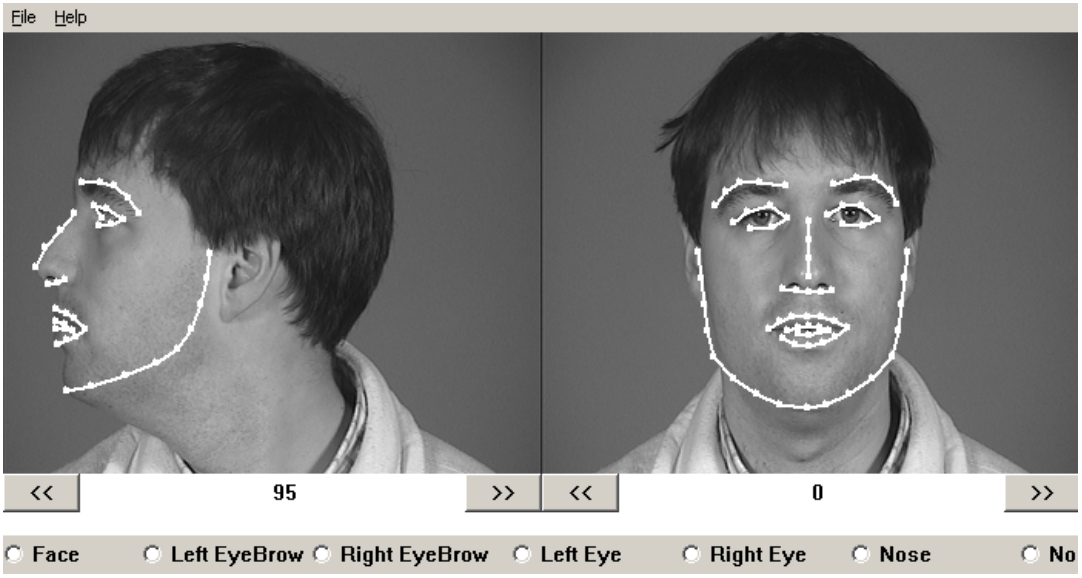


Figure 3.1: Landmark Placement

3.1.3 Texture Model and Parameters

After constructing 3D mean shape (as explained in the previous step), frontal texture of each facial image is warped into this mean shape based on the Delaunay triangulation as shown in the figure 3.3. This is the reason of calling it a 2.5D AAM, since it is composed of landmarks represented in 3D domain but only 2D texture is warped on this shape. Mean of these textures is calculated. Followed by another PCA to acquire texture parameters with 95% of the variation stored in these parameters.

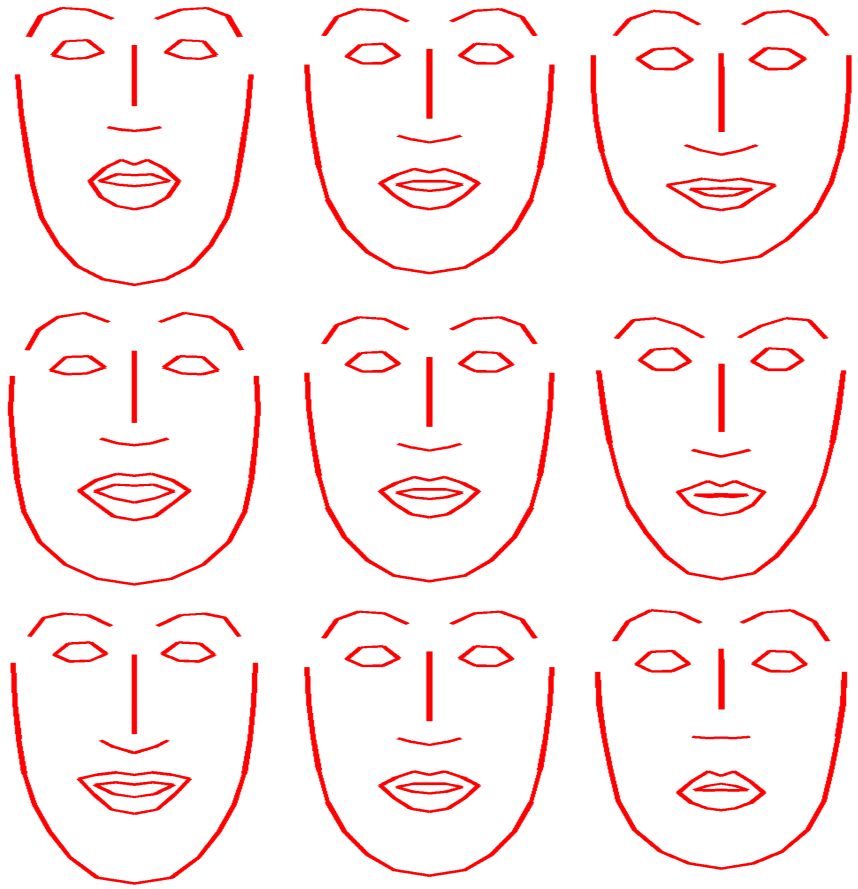
$$g_i = \bar{g} + \phi_g * b_g \tag{3.3}$$

where g_i is the synthesized texture, $\bar{g}$ is mean texture, ϕ_g is a matrix whose columns represents the eigen vectors obtained during PCA and b_g is a vector of texture parameters. Textures synthesized by incorporating first three texture parameters are shown in figure 3.4. All the textures corresponds to mean shape, therefore sometimes they are referred to as *shape-free textures*. Each column shows textures synthesized by varying respective texture parameter from $-3\sqrt{\lambda_i}$ (left), mean shape (center) and $+3\sqrt{\lambda_i}$ (right), where λ are the eigen values corresponding to each texture parameter (obtained during PCA) and i is the index of the texture parameter.

3.1.4 Appearance Model and Parameters

Both of the above parameters are combined by concatenation of b_s and b_g. And a final PCA is performed to have the appearance parameters.

$$b = [b_s b_g]^T, b = \phi_C * C \tag{3.4}$$



(a) $-3\sqrt{\lambda_i}$ (b) Mean Shape (c) $+3\sqrt{\lambda_i}$

Figure 3.2: Shapes synthesized by varying first three shape parameters from top to bottom

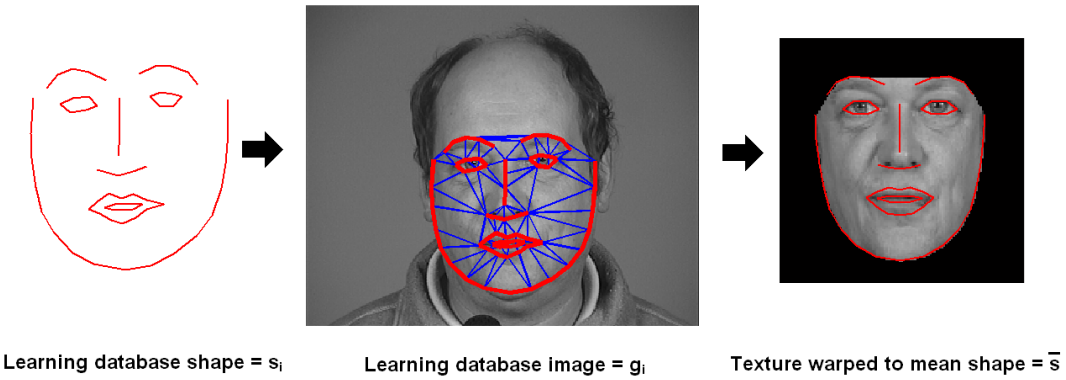
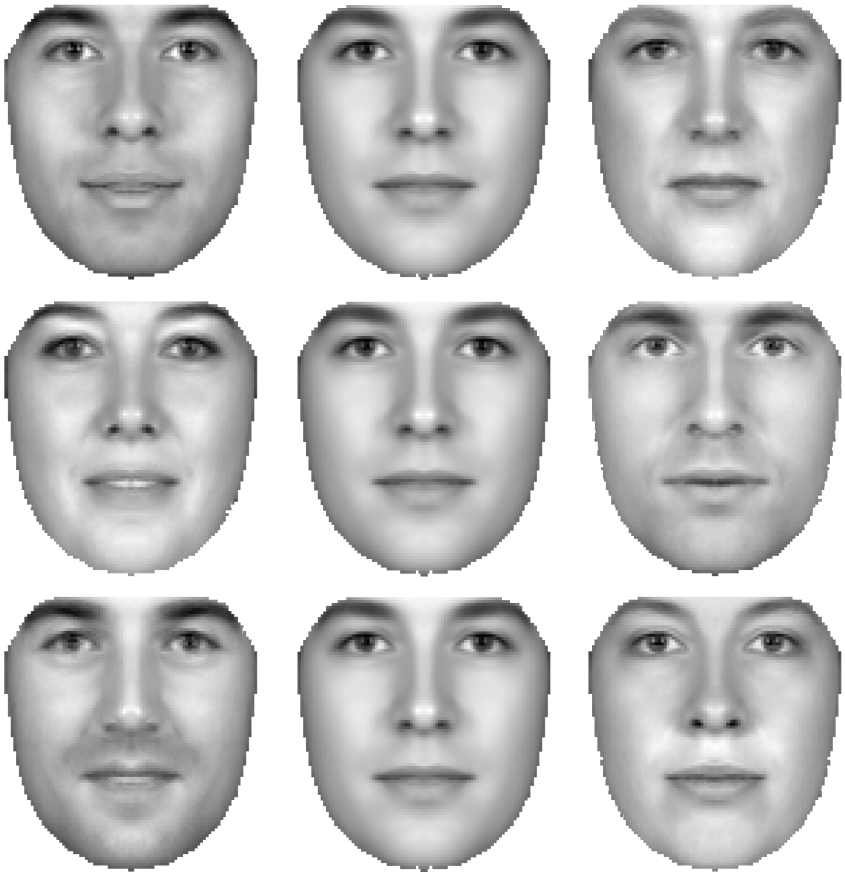


Figure 3.3: Texture Warping



(a) $-3\sqrt{\lambda_i}$ (b) Mean Texture (c) $+3\sqrt{\lambda_i}$

Figure 3.4: Textures synthesized by varying first three texture parameters from top to bottom. All of them are warped in the same mean shape.

where ϕ_c are the eigen vectors obtained by retaining 95% of the variation. And C is the matrix, whose column represents the appearance parameters of the faces of the database, which are used to obtain their shapes and textures. Model synthesized by incorporating first three appearance parameters are shown in figure 3.5. Since appearance model is a combination of a shape model and a texture model, therefore 3.5 appears to be a mixture of figures 3.4 and 3.2. Each column shows model synthesized by varying respective appearance parameter from $-3\sqrt{\lambda_i}$ (left), mean shape (center) and $+3\sqrt{\lambda_i}$ (right), where λ are the eigen values corresponding to each appearance parameter (obtained during PCA) and i is the index of the appearance parameter.

(a) $-3\sqrt{\lambda_i}$ (b) Mean Model (c) $+3\sqrt{\lambda_i}$

Figure 3.5: AAM Model synthesized by varying first three appearance parameters from top to bottom

3.1.5 Pose Parameters

2.5D Model can be translated as well as rotated with the help of translational and rotational parameters. Therefore we have three angles of rotation and three translational parameters named as pose parameters P.

$$P = [\theta_{pitch}, \theta_{yaw}, \theta_{roll}, t_x, t_y, scale]^T \tag{3.5}$$

where θ_{pitch} correspond to the face rotating around x axis (shaking head up and down), θ_{yaw} to the face rotating around y axis (profile views) and θ_{roll} to the face rotating around z axis. t_x, t_y are the offset values from the face center detected by a face detector and *scale* is the magnification of the model. Figure 3.6 shows the face rotating around y axis making left and right semi profile views.

Figure 3.6: Snapshots of rotating 2.5D AAM

In segmentation phase shape of this 2.5D AAM can be deformed, rotated and translated by varying C and P parameters. With the help of this shape, face in the query image I is warped to the frontal view. The objective is to minimize pixel error between this warped query image I and AAM model obtained by C parameters.

$$e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[I_i(C,P) - M_i(C)]^2} \tag{3.6}$$

where $I(C,P)$ is the segmented image and $M(C)$ is the model obtained by C parameters and N is the number of pixels of the model.

3.2 Facial Image Databases

2.5D AAM model discussed in previous section requires an adequate amount of facial images to learn the variations in different classes of faces. Similarly to validate the facial analysis system a sufficient number of test facial images are required. These facial images databases are generally referred to as learning databases and testing databases. This section will discuss and present both databases in detail.

3.2.1 Learning Database

As explained in section 3.1, to build the proposed 2.5D AAM model frontal and profile views of a face are required. M2VTS database of [94] is chosen as a learning database in

this thesis work. Its high resolution, constant illumination, occlusion free facial images makes M2VTS most suitable database to be used for 2.5D AAM modeling.

M2VTS M2VTS (Multi Modal Verification for Teleservices and Security applications) database is made up from 37 different faces rotating the head from 0 to -90 degrees, again to 0, then to +90 and back to 0 degrees. The sequences are meant for facial analysis and provide information about the 3-D face features. Along with face type variation, it also includes variations with respect to head position, eyes opened/closed, different hairstyle and faces with beard. A Hi8 video camera was chosen for the shooting. By keeping active pixels only, the final resolution for the database images is 286x350 pixels. The database can be considered as having been produced under "ideal" shooting conditions (good picture quality, indoor shooting, nearly constant lighting, uniform gray background) and within a highly co-operative scenario (as much as they could, people followed the instructions they were given).

2.5D AAM model is build by taking the shape model of both frontal and profile of all the 37 faces as shown in 3.1, whereas only frontal view textures for texture parameters. Some of the images of this database is shown in 3.7.

Figure 3.7: M2VTS: Some examples of learning database

3.2.2 Test Databases

To validate the performance of facial analysis system, proposed algorithms are tested and compared with respect to various test facial image databases. Since algorithms under observation are tested both in single camera and multiple cameras system therefore both types of databases are acquired to perform simulations. Among them single camera databases are easy to obtain due to the enormous amount of databases available in the research community, whereas for scarcely available multiple camera images, a multi-view scenario is implemented.

3.2.2.1 Single Camera Databases

Proposed algorithms of facial analysis system are tested and compared with the help of the following facial images databases.

Pointing'04 The head pose database of Pointing'04 [18] used in this thesis consists of facial images of 15 individuals. Two sets of 13 facial images of each individual moving their faces laterally from -90 degrees to +90 degrees are taken. To obtain different poses, markers have been placed in the whole room. Each marker corresponds to different yaw angle of head i.e. $\pm 15°$, $\pm 30°$, $\pm 45°$, $\pm 60°$, $\pm 75°$ and $\pm 90°$. In order to obtain the face in the center of the image, the person is asked to adjust the chair to see the device in front of him. After this initialization phase, each person is asked to stare successively at points without moving his eyes. Sequence of facial images one of the individual of Pointing'04 is shown in 3.8.

Figure 3.8: Some examples of POINTING'04 database

SUPELEC SUPELEC facial database consists of 7 individuals (of SCEE team) making lateral facial movements from -90 degrees to + 90 degrees in front of a webcam. The camera is placed at 70 cm distance from the individual and a constant ambient light is used to acquire 218 images of the database. Figure 3.9 shows some examples of this database.

Synthetic In previous two databases of SUPELEC and POINTING'04, facial orientation of individuals are not precise and it is difficult to ask somebody to maintain specific roll, pitch and yaw angles (e.g. in the first and last images of the third row in figure 3.9, individual was unable to maintain constant pitch angel). In order to compare the results of the algorithms, precise values of ground truths of facial orientation are required.

Figure 3.9: Some examples of SUPELEC database

Therefore, scenario of SUPELEC database is emulated in a commercially available software called MAYA to acquire the synthetic facial images with high precision of facial orientation. Figure 3.10 shows some examples of this database.

These facial images are from two sources. Some of them are indigenous, while some of them are build in a software called Facial Studio. In Facial Studio one can generate various synthetic faces by varying its appearance with respect to ethnicity, gender and by changing location, width, height and textures of its facial features. Sixty faces were generated from facial studio (last row of figure 3.10)and two others are indigenous (first two rows of figure 3.10). Each of the face is imported in MAYA and is rendered in the form of 120 images while moving laterally from $-90°$ to $+90°$ and back to $-90°$, such that each image has a change of $3°$ of yaw angle while other translational, rotational and appearance parameters remains constant.

3.2.2.2 Multiple Camera Databases

Multi Camera System Setup Database of face image capable of self assessing is desired to validate multi-view facial analysis system. The community lacks database which involve lateral motion of a face captured by more than one camera. In order to implement the system a multi-view scenario has been developed shown in figure 3.11. The purpose of constructing this multi-view system is to emulate the scenario of integrating two off the shelf webcams placed on the extreme edges of display screen facing towards the user as shown in figure 3.11.

These cameras are placed 50 degrees apart on a boundary of a circle with a radius of 70 cm. Center of this circle represents the "look at" point for each camera. Third camera is also placed between these two cameras, which is not the part of the multi-view system but placed for the evaluations and comparisons of single-view and multiple-view algorithms.

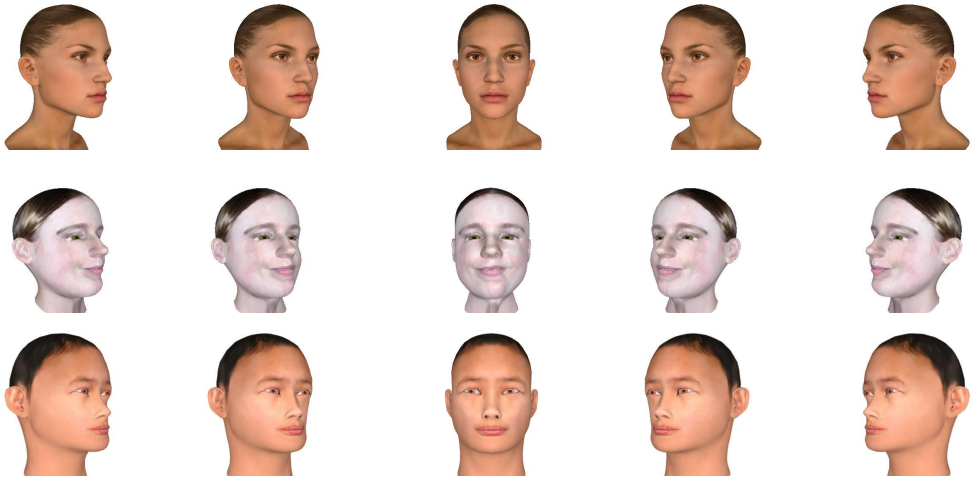


Figure 3.10: Some examples of Synthetic database

Calibration of cameras is performed by a publicly available toolbox [95]. Both intrinsic and extrinsic parameters are calculated and all the images of these cameras are calibrated.

SUPELEC Seven individuals of a research team are invited for screen shots with the intention of obtaining 1218 images with lateral motion. Each individual rotates his face gradually from frontal view to left and right profile views. At each instance three images from each webcam are acquired simultaneously to obtain temporally synchronized images. For the steady state illumination a white ambient light is placed behind the central camera. Illumination remains steady through out the sequence. Figure 3.12 shows some images of testing database acquired from three webcams.

Synthetic Similar scenario is emulated on software MAYA for a video of synthetic face. Synthetic face database does not contain camera calibration error hence it is helpful to analyze results free of calibration errors. 4160 facial images (from each camera) of 52 synthetic characters are acquired by the software MAYA as explained in the previous section 3.2.2.1. Figure 3.13 shows some examples of testing database of synthetic face.

3.3 Ground Truth Error (GTE)

As explained in section 1.1.2, pose estimation of an unknown face is one of the problem stated in this thesis. As the user is not obliged to remain in front of the camera therefore facial pose estimation becomes necessary for a Cognitive Radio equipment. The precise value of the facial orientation is unachievable in real faces, therefore its accurate comparison is carried out only in synthetic face databases. While for real faces a snapshot of the results are shown to have an idea of the accuracy of the results.

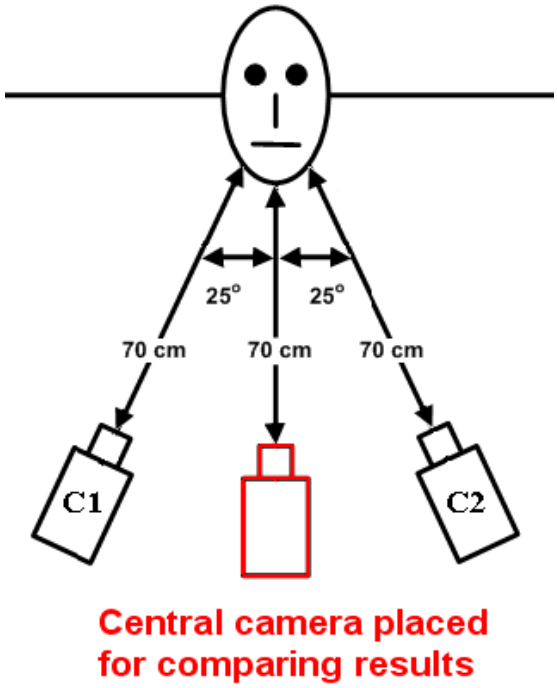


Figure 3.11: Multi-View System

Similarly in section 1.1.2, problem of localization of the facial features of an unknown face is stated. CR equipment required the exact location of these features in order to perform their lossless compression, unlike remaining of the facial region. Therefore a method, called GTE, is required for the comparison of the accuracy of the localization of these features for different algorithms. This section provides the detailed procedure for calculating GTE.

Shape model and localization of features obtained by the proposed algorithms are compared with respect to the key features of the face i.e. center of eyes, tip of nose and center of gravity of the mouth. For this comparison, four points of ground truth (center of eyes, tip of nose and center of mouth) are marked manually on each facial image of all the test databases discussed above.

Mean of error distance (Euclidean distance) between ground truth points marked manually and the points given by the shape model obtained by the experiments is calculated. This error is normalized by D_{face} (distance between center of the mouth and the line joining the center of the eyes) to acquire Ground Truth Error (GTE). In the community D_{eye} (distance between center of eyes) is taken for the normalization of GTE, however in our case face rotation causes variations in D_{eye}. To compensate it we take D_{face} and find its equivalent D_{eye} i.e. $D_{eye} = 0.8 * D_{face}$ to normalize Ground Truth Error (GTE). Therefore GTE is expressed as a percentage of the distance between the eyes, i.e. an error of 1 corresponds to a mean error equal to the distance between the eyes. It is given as

Figure 3.12: Examples of real webcam facial images of multi-view system (Same pose from 3 webcams)

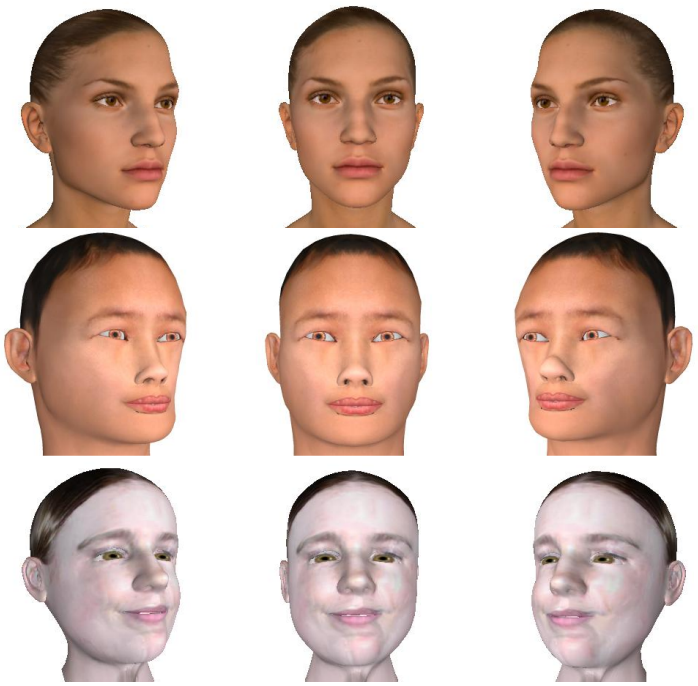


Figure 3.13: Examples of synthetic facial images of multi-view system (Same pose from 3 cameras)

$$Error = \frac{1}{4D_{eye}} \sum_{i=1}^{4} d_i \tag{3.7}$$

where d_i corresponds to error distance for each feature. Due to the fact that ground truth points are marked manually one needs to define a minimum threshold which eliminates this vagueness. Any two algorithms having a percentage of GTE less than 10% of D_{eye} is considered to be equally accurate. While for the maximum threshold results less than 25% of D_{eye} is considered to be well converged results. Therefore the results are compared for GTE greater than 10% and less than 25% of D_{eye}. Figure 3.14 shows the range of minimum and maximum thresholds of GTE.

Figure 3.14: Circular regions around facial features of each image, from left to right, represents 25%, 20%, 15% and 10% of GTE

3.3.1 GTE for Multiple Camera Images

GTE calculation for single view images are simple whereas in a multi-view system, since facial images from all the cameras are directly involved in the segmentation phase of the 2.5D AAM and GTE obtained from each image are equally important, therefore it is not appropriate to calculate GTE only for the images obtained from any one of the camera. Taking mean of these GTEs is not a suitable solution. In this thesis a unique solution of calculating GTE is proposed by using an extra (third) camera.

In a multi-view system, third camera is placed in between the two cameras as shown in figure 3.11. Images from this camera is not used in the segmentation phase and only used to calculate GTE. The coordinates of this camera are calculated while performing the calibration of multi-view system. AAM model obtained at the end of the facial analysis of an image from other two cameras is rotated and translated with respect to the coordinates of this third camera. Finally GTE of AAM model with respect to the facial image seen by this camera is calculated for the comparison of the methods. With the help of this camera, single camera test database is also obtained as explained in section 3.2.2.1, therefore facial analysis results by single-view system can also be compared with those by multi-view system on same facial images. Moreover tedious job of annotating four ground truth points on hundreds of facial images of multiple cameras is also reduced effectively. Similarly for any other configuration of multiple cameras system, an extra camera can be placed elsewhere whose images are used only for GTE calculation and not in AAM segmentation phase.

3.4 Conclusions

This chapter has presented in detail the construction of newly proposed 2.5D AAM model. Main advantages of this model compared to other 3D AAM models are i) more practical compared to 3D scanner models ii) key facial features captures by few points iii) storage of only frontal view texture in memory and iv) requires only two facial images (frontal and profile) of the individual instead of applying triangulation on various 2D images. The only disadvantage of this model is the lack of the depth data around regions without markers e.g. cheeks.

This chapter has also presented facial image databases both for single and multiple camera system used throughout in this thesis. Moreover it also discusses the way of calculating the ground truth error of facial features localization for the comparison of different algorithms which will be discussed in detail in subsequent chapters.

Next two chapters will present our remaining two contributions of the application of this 2.5D AAM by

- for single camera system: Hybrid GA+GD optimization for AAM.
- for multi-view system: Hybrid multi-objective GA+GD optimization for AAM.

Chapter 4

AAM fitting for Single View Images

Contents

The problem stated in this thesis for the facial analysis i.e. "pose estimation and facial features localization of unknown and oriented faces", is dealt with in this chapter for single camera configuration. It presents our second contribution of an efficient optimization technique for AAM by the hybridization of genetic algorithm (GA) with gradient descent (GD) to make a robust, efficient and real time face alignment system. Facial large lateral movements requires to optimize 6DOF (Degrees of Freedom) pose parameters which make the facial search space of AAM non-convex. This non-convex multidimensional search space requires an efficient optimization methodology. Section 4.1 discusses in detail the optimization methods used for face search by AAM. Section 4.2 discusses in detail the hybridization of these methods proposed by the community in different domains. We select two hybrid algorithms of GA-Simplex and GAGD for the implementation in AAM. In a non-convex search space GA is able search face globally whereas GD, also known as deterministic optimization method, can search face locally.

As far as simplex is concerned it is an intermediary approach for face search. Therefore exploitation properties of GD and simplex are combined with the exploration property of GA in GA-Simplex and GAGD respectively.

Simple GA-Simplex hybridization is carried out by separately applying GA and simplex in series. Whereas for GAGD we propose a gradient operator in GA in section 4.3, which functions in conjunction with the existing genetic operator of mutation. Thus it does not increase the computational cost of the system. Stepwise application of our proposed algorithm is also explained in this section. We compare it with classical search algorithm by gradient descent and the hybrid optimization of GA with Simplex in section 4.4. Facial databases of SUPELEC'08, Pointing'04 and synthetic characters comprising of different facial poses are analyzed. Simulation results validate the efficiency, accuracy and robustness achieved.

4.1 Optimization Methods for AAM

Optimization methods plays an important role in the facial analysis by AAM. System's efficiency and robustness is directly related to the optimization method for the facial search in segmentation phase.

As discussed earlier in previous chapters, classical optimization of AAM by linear regression presents some drawbacks. It needs to store the matrices R_C and R_P in memory for the segmentation phase. Additionally the precomputations from the training set is only an approximation for any given target image, and may be a poor one if the target image is significantly different (unknown or unseen faces) from the training images. This capability of analyzing the unseen faces is called generality and is investigated in the table 4.1 for different optimization techniques of AAM discussed in this section. This thesis work is going to resolve the problem of generalization by working on the optimization technique. AAM modelization used M2VTS database, which is sufficiently general to align the unknown faces, but the face search optimization technique needs to explore the search space for these unknown faces. Therefore in this thesis we are going to address the problem of generality by concentrating on the efficient and robust optimization technique for AAM.

In segmentation phase the deformed, rotated and translated shape model obtained by varying C and P parameters of 2.5D AAM (obtained in the previous chapter), is placed on the query image I to warp the face to mean frontal shape. The objective is to minimize pixel error

$$e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[I_i(C,P) - M_i(C)]^2} \tag{4.1}$$

where $I(C,P)$ is the segmented image, $M(C)$ is the model obtained by C parameters and N is the number of pixels. The objective of pixel error minimization is obtained by optimizing C and P parameters, which requires an efficient and robust optimiza-

tion technique. In general, optimization methods can be divided into two categories; deterministic methods and direct search methods.

Deterministic methods can be defined as mathematical techniques based on the concept that future behavior can be predicted precisely from the past behavior of a set of data. These methods ignore the existence of disturbances that may alter the data's future pattern. e.g. Gradient descent.

Direct search methods are nonlinear optimization methods that neither require nor explicitly approximate derivatives for the problem to be solved. Instead, at each iteration a set of trial points is generated and their function values are compared with the best solution previously obtained. This information is then used to determine the next set of trial points. e.g. Genetic Algorithm, Tabu search.

Separation between these two methods is not strict and various methods can be placed in between e.g. Nelder Mead Simplex. Main reason of separating them is their properties of exploitation and exploration. One of the main advantages of direct search methods over deterministic methods is the parallel processing. Since direct search methods do not predict from past behaviors, therefore they can perform processing in parallel in hardwares like Graphical Processing Unit (GPU). Detailed discussion of some of the well known methods and their implementation in AAM are presented in the next subsections.

4.1.1 Gradient Descent

Gradient descent method is a well known method which finds the optimum value of an error by calculating the gradient of respective function with respect to each parameter. It is calculated as a difference between segmented image and the model instance by changing each of the parameter C and P independently. It is given as

$$\frac{\partial e_x}{\partial C_i} = \frac{(I(C_{i,2}, P) - M(C_{i,2})) - (I(C_{i,1}, P) - M(C_{i,1}))}{C_{i,2} - C_{i,1}} \tag{4.2}$$

$$\frac{\partial e_x}{\partial P_i} = \frac{(I(C, P_{i,2}) - M(C)) - (I(C, P_{i,1}) - M(C))}{P_{i,2} - P_{i,1}} \tag{4.3}$$

where i indicates index of C and P parameters. e_x represents error image or residual error (of the same dimensions as of M) and x corresponds to the number of pixels. Each pixel of the error image e_x corresponds to the pixel error of a particular pixels of I and M, unlike equation 4.1 which represents the root mean square of these pixels' errors. Gradients with respect to each parameter can be considered as partial derivatives of a function. Matrix obtained by calculating these partial derivatives of error images is called Jacobian matrix. Jacobian matrices of C and P parameters are given as

$$J_C = \begin{pmatrix} \frac{\partial e_1}{\partial C_1} & \frac{\partial e_1}{\partial C_2} & \cdots & \frac{\partial e_1}{\partial C_M} \\ \frac{\partial e_2}{\partial C_1} & \frac{\partial e_2}{\partial C_2} & \cdots & \frac{\partial e_2}{\partial C_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N}{\partial C_1} & \frac{\partial e_N}{\partial C_2} & \cdots & \frac{\partial e_N}{\partial C_M} \end{pmatrix} \tag{4.4}$$

$$J_P = \begin{pmatrix} \frac{\partial e_1}{\partial P_1} & \frac{\partial e_1}{\partial P_2} & \cdots & \frac{\partial e_1}{\partial P_6} \\ \frac{\partial e_2}{\partial P_1} & \frac{\partial e_2}{\partial P_2} & \cdots & \frac{\partial e_2}{\partial P_6} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N}{\partial P_1} & \frac{\partial e_N}{\partial P_2} & \cdots & \frac{\partial e_N}{\partial P_6} \end{pmatrix} \tag{4.5}$$

where N is the number of pixels and M is the number of appearance parameters. These Jacobian matrices of parameters direct their respective parameter to optimum solution by pointing to the variation of each parameter. Hence ΔC and ΔP are calculated by multiplying the transpose of respective Jacobian matrix with an error image obtained at any instance.

$$\Delta C = -\eta \frac{J_C^T}{J_C^T J_C} e_x \tag{4.6}$$

$$\Delta P = -\eta \frac{J_P^T}{J_P^T J_P} e_x \tag{4.7}$$

where η is the step size to control the jump of parameters in the direction of the gradient.

Gradient based methods are known to lack exploration property hence usually fall into local minimum. As shown in figure 4.1, the pixel error curve by changing θ_{yaw} and t_x of the model, introduces several local minima. Initializations around every local minimum can lead to better convergence by these methods. Similar initializations are also required for other pose parameters. Thus the numbers of initializations are so huge that it is impractical to use these methods. However instead of these initializations, it is far better to use genetic algorithm which explores the error curve with out falling into the local minimum.

4.1.2 Genetic Algorithm

Genetic algorithms (GA) categorized as an evolutionary algorithm. Idea of evolutionary computing was introduced in the 1970s by I. Rechenberg in his work "Evolution strategies" [96]. His idea was then developed by other researchers. Genetic Algorithms (GAs) were introduced by John Holland's book "Adaption in Natural and Artificial Systems" [97]. GAs are often used for global optimization problems, therefore they can search face globally even in scattered face search space. We have used it to optimize C and P parameters of AAM. All the parameters (genes) of C and P are concatenated to form a chromosome as shown in the figure 4.2.

Specific numbers of chromosomes are initialized to make a population, where each chromosome corresponds to a model instance. Pixel errors (fitness) is calculated between

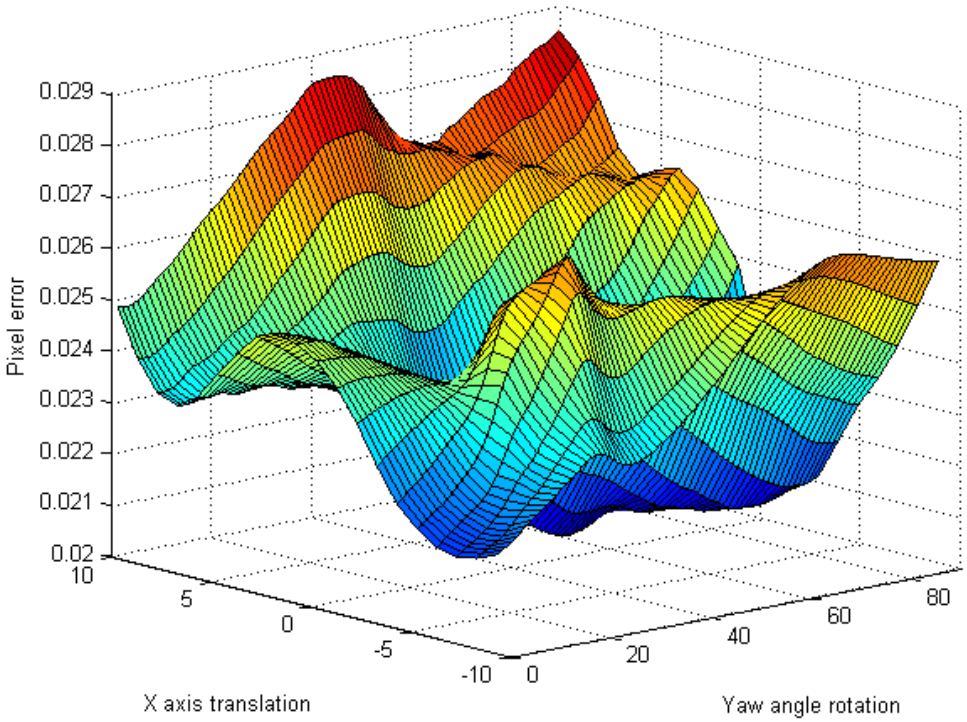


Figure 4.1: Pixel Error by changing t_x and θ_{yaw}

the warped query image and AAM model ($M_{C,P}$) represented by a single chromosome by equation 4.1. Tournament selection is applied to select parents (best fit from a pool of randomly selected parents) from the population to undergo reproduction. Two points crossover and Gaussian mutation are implemented to reproduce next generation of the chromosomes (shown in figure 4.3) and ultimately replacing old ones. In this way new generation of the same size of population is created using genes of the fittest of the old chromosome. Elitism is also implemented to preserve the best possible solution at all time. After several generations most of the chromosomes surrounds the optimum solution, but most of the time none of them bear the global optimum.

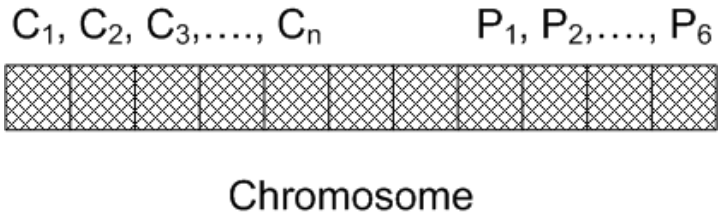


Figure 4.2: Chromosome

GA is renowned to have a good quality of exploration; as a result it can find the global optimum values. It is an iterative and population based method. In an iteration of a classical GA, a new generation of population of chromosomes is evolved based on genetic operators of crossover and mutation. Crossover operator exchanges the genes (parameters) of the selected parents to form child chromosomes, thus it makes an effort to exploit (to some extent) the existing solutions to produce a new better solution. Whereas a mutation operator mutates genes of a chromosome and ultimately explores the search space without any preferred direction. The rate of the probability of executions of these operators in an iteration makes GA either highly exploratory (due to high mutation) or slightly exploitive (due to high crossover). Thus GA propose a good compromise between exploitation and exploration.

4.1.3 Simplex

Nelder Mead Simplex algorithm proposed by Nelder and Mead [86] can also be used to optimize the pose and appearance parameters of AAM. The target is to find out the best possible value of these parameters giving minimum pixel error (equation 4.1) between model and the query facial image. Simplex is an iterative and population based algorithm. It starts with a population of N+1 (where N is the number of parameters) solutions initialized randomly. In each iteration a new solution is calculated and inserted in the population by applying simplex operators (reflection, expansion, contraction or shrunk [86]) on the existing population. Followed by the sorting of the population with respect to pixel errors and dropping off the worst solution in order to keep the population size constant. It converges when no further good solutions are possible. For stopping criterion fixed number of these iterations is applied, which is either fixed by

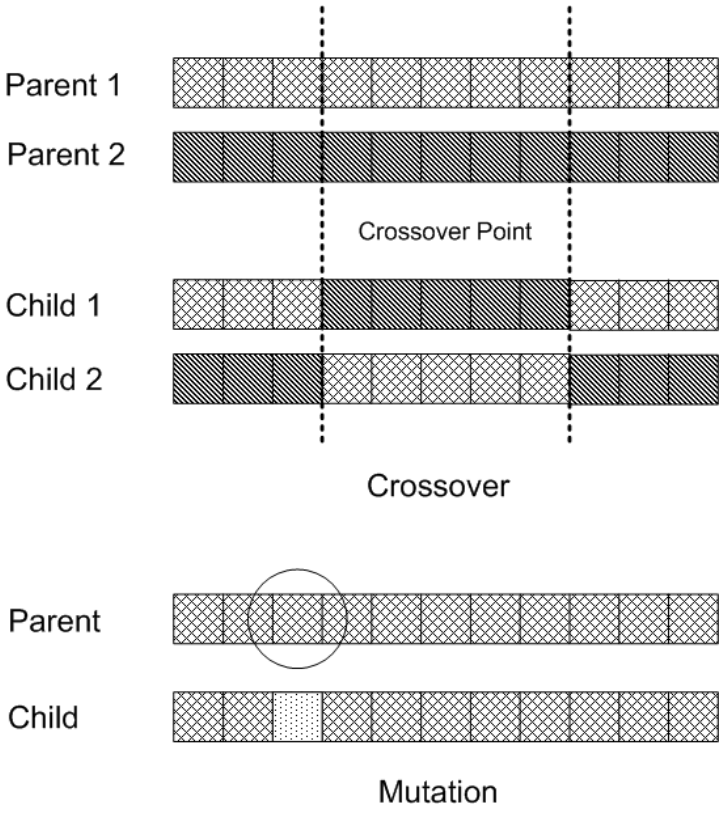


Figure 4.3: Crossover and mutation operator

number of convergences or processing time. In the end it provides a set of solutions, in which best solution is the one with least pixel error.

Simplex is considered to be an intermediary approach for both exploration and exploitation. Although as GD it does not have a direction in the form of a gradient to find good solutions but its previous solutions in the population assist it to find this direction to some extent. Therefore its convergence to local optimum is slower than GD. On the other hand, operators like expansion and reflection tends it to explore the solutions beyond its direction of convergence and eventually make it a better exploratory method. However another drawback of this algorithm which makes it less exploratory than GA is the inability of insertion of solutions worst than the current population. Although modifications can be done to improve its exploitation and exploration, but generally simplex is considered to be an intermediate approach between GA (highly exploratory) and GD (highly exploitive).

4.1.4 Other Methods

Various other optimization methods exist which are suitable for the non-convex optimization problems. Following methods are discussed briefly in this section: Random Search/Random Walk, Simulated Annealing, Monte Carlo and Tabu search. Since these methods are never used in AAM, therefore their functionalities are discussed with respect to general optimization problem.

A random search (RS) is a simplest search strategy, as it simply evaluates a given number of randomly selected solutions. A random walk (RW) is very similar, except that the next solution evaluated is randomly selected using the last evaluated solution as a starting point [98]. These strategies are not efficient for the optimization problems of AAM.

Simulated Annealing (SA) is an algorithm explicitly modeled on an annealing analogy [99]. SA borrows ideas from a physical procedure called annealing where a substance is melted and then slowly cooled down in search of a low energy configuration. In a similar manner, probabilistic optimization is performed with a decreasing temperature that determines how greedy the procedure is in the search for a global minimum. It is an iterative process and at each step, the SA considers some neighboring solution of the current solution, and probabilistically decides whether to stay or move to that solution. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Typically this step is repeated until the system reaches a solution that is good enough for the application, or until a given computation budget has been exhausted. The probability of making the transition is specified by an acceptance probability function that depends on the two solutions and on a global time-varying parameter T called the temperature.

In general, Monte Carlo (MC) methods employ a pure random search where any selected trial solution is fully independent of any previous choice and its outcome [100]. The current "best" solution and associated decision variables are stored as a comparator. Tabu search (TS) is a meta-strategy developed to avoid getting "stuck" on local optima. It keeps a record of both visited solutions and the "path" in the form of a tabu list.

This information restrict the choice of solutions to evaluate next.

All these methods are although suitable for non-convex problems and are good in exploration as well, but nobody has used them with AAM because of their slow rate of convergence to optimum value.

All the method discussed above have their utility in their respective domains, where the system may require either highly exploitive or highly exploratory optimization. From the detailed discussion in the above sections we can conclude the characteristics of the algorithms as tabulated in table 4.1. It illustrates that SA, TS and GA are highly exploratory method with little exploitation. Whereas GD is a highly exploitive method with little exploration. In this table the previously discussed generality is also compared to AAM optimization methods using precomputation. In the next section we will present their hybridization in detail.

Method	**Exploration**	**Exploitation**	**Generality**
With Precomputation	+	+++	+
RS/RW, MC	+++	-	+++
SA, TS, GA	+++	+	+++
Simplex	++	++	++
GD	+	+++	++

Table 4.1: Characteristics of Optimization Algorithms

4.2 Hybridization

From the above discussion we have presented the exploitation and exploration properties of various methods. Since in face analysis the multidimensional search space formed by AAM is non-convex, therefore an optimization methods comprising of both the properties is required. Utilizing any one of the algorithm of table 4.1 alone, will not solve the problem. One other way is to hybridize these algorithms. In this section first of all we will discuss some of the hybrid algorithms presented by various authors in their respective domains in section 4.2.1. Followed by the discussion on two hybrid algorithms of genetic algorithm with Simplex and genetic algorithm with gradient descent.

4.2.1 Previous Work

Smart and Zhang [101] used gradient descent search to genetic programming for object recognition. During the evolutionary process, the search is based on a global search mechanism, but the gradient descent search is locally applied to individual programs in the population inside a particular generation. They applied GD inside a particular generation, hence making system slower. Fernandez et al. [102] proposed a single solution instantaneous memetic algorithm for the correction of illumination in homogeneities in images. They replaced conventional mutation operator, by mutating the solution using the gradient information of the solution. Therefore they calculated gradient of

the solution in each iteration for their new mutation operator, thus causes the system to become slower. Alker et al. [103] extracted 3D landmark of MR and CT images of human head by hybridizing GA with conjugate gradient optimization. Unfortunately, conjugate gradient is computationally very expensive, due to the recalculation of the search direction at each iteration. Similar to Quasi-Newton method which requires extensive computation of the Hessian matrix (second-order derivatives).

Bosman and Thierens [104] exploited gradient information in IDEA (Iterated Density Estimation Evolutionary Algorithms) to optimize some well known difficult continuous differentiable functions. They applied gradient descent in each iteration on randomly selected solutions from the population. They applied their proposition on Griewank's, Michalewicz's and Rosenbrock's functions. Main drawback of their method is the lack of intelligence of applying GD i.e. when and on which solutions of the population, GD should be applied. Applying GD on randomly chosen solutions at the end of each iteration makes the system slow.

Durand and Alliot [5] combined Simplex with GA and tested on classical test function of Griewank and Corona. Skinner et al. [105] used iterative two stage hybrid optimization of parallel GA followed by Sequential Quadratic Programming (SQP). They tested the algorithm on classical mathematical functions. Applying SQP after the GA optimization is time consuming. Moreover their testing methods are not related to the problem stated in this thesis.

Zhang and Ma [106] developed an efficient hybrid GA for continuous optimization problem. They inserted a local search method (LSM) in the crossover operator to find better offsprings. Their proposed LSM tried different combination of genes in a parent chromosome during crossover and finds the best combination. Introducing these best chromosomes into the population makes the overall GA more exploitive, but this procedure of local search is very time consuming. Zdansky and Pozivil [107] combined GA with Tabu search to optimize scheduling of flowshops of plants in a chemical industry. They used tabu search as a local improvement technique, which is also able to leave local optimum and continue the search. In their method they initially apply tabu search on the whole population and then apply GA on a smaller set of these solutions to make a new generation. Their methodology also seems slower because of the combination of two highly exploratory algorithms.

Although all the experiments by these hybrid evolutionary algorithms showed an improvement over the conventional optimization, but their utility in a real time system is not practical. Their way of combining two algorithms are no doubt robust but are not time efficient compared to our proposition. The hybrid algorithms selected for this thesis are

Hybrid GA-Simplex Both GA and simplex are population based algorithm, therefore their hybridization is carried out by transferring the set of solutions given by GA after the stopping criterion is met. Simplex takes this population as its initialization and starts the search for the best solution. This hybridization has been tested on various mathematical functions in [5], almost similar approach will be carried out by us in this thesis. Main objective of implementing this combination is to verify the robustness

of our proposed following algorithm.

Hybrid GA-GD Hybridization of GA and GD is not that simple compared to GA-Simplex, due to the different nature of the two algorithms. Various authors have combined these algorithms in different manner as presented in the previous section, but their propositions are not time efficient. Although they have embedded GD in GA, but their propositions are meant to replace GA exploration by GD exploitation or vice versa. Whereas the better way is to hybridize them without altering their indigenous qualities. Next section discusses our unique way of this hybridization for AAM called HGOAAM.

4.3 Hybrid Genetic Optimization for AAM (HGOAAM)

A simple hybridization of gradient descent in GA would have increased the number of error evaluations, but we propose *gradient operator* which functions in conjunction with mutation operator. This gradient operator uses the error evaluation of mutation operator and do not put an extra burden on the system. This section presents our contribution of hybridizing GD with GA by gradient operator followed by the stepwise explanation of our proposed hybrid AAM.

4.3.1 Gradient Operator

During mutation of a parent chromosome we have changed only one gene or parameter to make a child for the next generation. This property of the mutation operator enables us to retrieve the residual image (pixel error) as presented by e_x in equations 4.2 and 4.3, with respect to each C and P parameter. These errors are actually partial differential of the error with respect to each parameter. During generation evaluation whenever a chromosome undergoes mutation, the gradient operator stores these partial differential of respective parameter to build Jacobian matrices of equations 4.4 and 4.5 without interrupting the mutation operator's functionality.

In the beginning of GA evolution, these Jacobian matrices are not accurate as the search space is scattered. But as the GA evolves for few generations it surrounds the region around the global optimum. At this time Jacobian matrices calculation is worthwhile. Figure 4.1 is view from different angles in figure 4.4 to point out these global and local minima.

Separate experiments were conducted to verify the stability of the Jacobian during the GA evolutions. In these experiments GA was executed normally and meanwhile gradient operator calculates Jacobian for mutated genes in each generation. These Jacobian are used to calculate the expected value of the gene (parameter) pointed by the gradient of the Jacobian. Figures 4.5 show the first three C parameters, θ_{yaw}, t_x and t_y evaluated by the Jacobian in each generation. From these figures we can see that parameters optimized by Jacobian become stable i.e. point in the right direction after 7 to 10 generations of GA evolution, before that the evaluation of parameters by

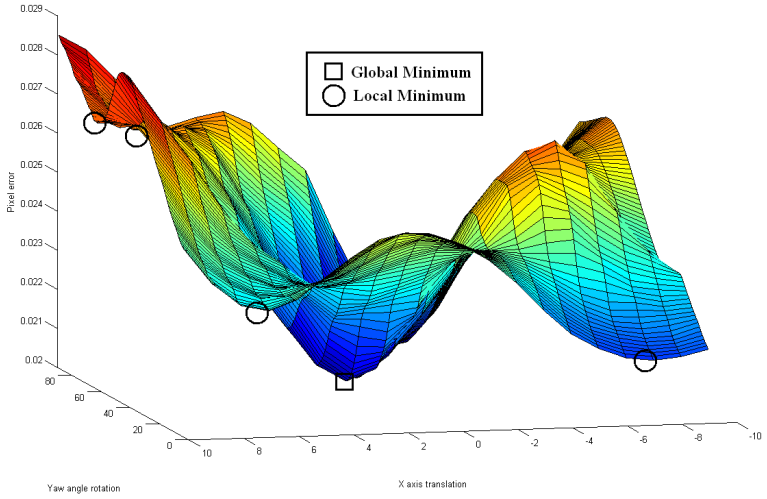


Figure 4.4: Representation of global minimum region surrounded by local minima.

Jacobian were erroneous. Which means that the parameters' solutions have reached the region of global optimum value.

Therefore after $b = 10$ generations of our experiments, Jacobian matrices with respect to each parameter are calculated in all the subsequent generations. Until unless all the parameters of the chromosomes are not mutated and operated by the gradient operator, complete mean Jacobian matrices can not be evaluated. After completion, these Jacobian matrices are applied with the help of equations 4.6 and 4.7 on half of the population, which was used to get copied unaltered from previous generation, to obtain best children of these parents. Applying this operator only on half of the population allows us to explore and exploit the error function at the same time i.e. this operator keep the individual qualities of GA and GD intact. Even if this half of the population has fallen in the local minima, still parameter responsible for this fall can help to find global minimum in the subsequent generations of genetic algorithm, where genes get exchanged in a chromosome through crossover operator.

Let us consider there are N number of chromosomes in a population, P_m is the gene wise mutation probability and P_k chromosomes undergoes mutation in a single evolution. Then the probability of calculating gradient of each parameter in a single generation is

$$P_{J_{param}} = P_m * P_k * N \tag{4.8}$$

Frequency of application of this operator on the population depends on the probability of occurrence of mutation, gene wise mutation and number of chromosomes. Experiments showed that by keeping the mutation and its gene wise probability to 20% and 2% respectively, gradient operator requires at least five generations to completely calculate the Jacobian Matrices (i.e. $P_{J_{param}} \geq 1$) for a population of 50 chromosomes. Hence this operator will not introduce significant time delay since it retrieves

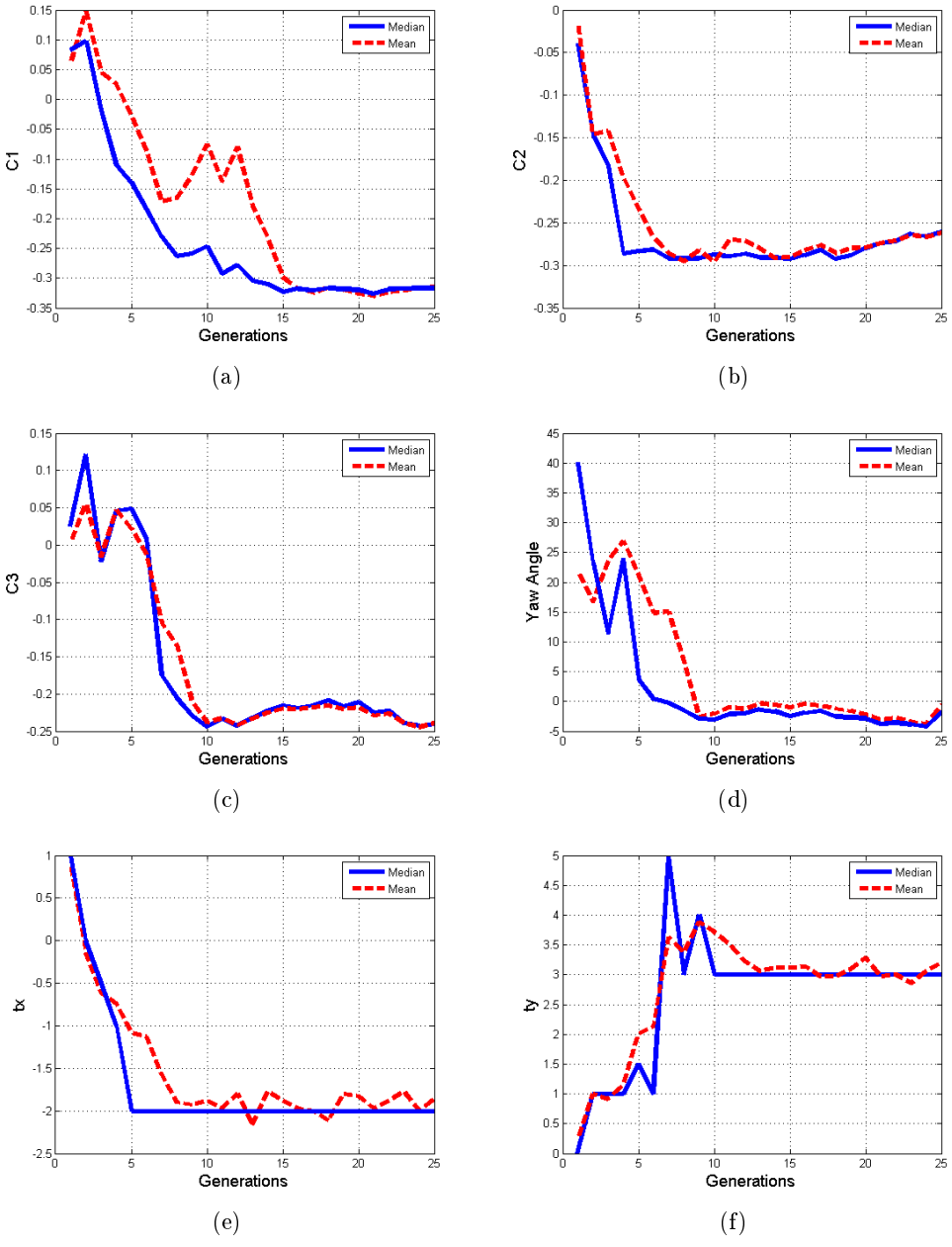


Figure 4.5: Evaluation of some of the C and P parameters by Jacobian with respect to each generation

information from another operator and it is applied for fewtimes after evolution of 10 generations of GA alone. Additionally, Jacobian matrices are not that huge to store while evaluating solutions.

4.3.2 HGOAAM Fitting

This section presents the pseudo code and the stepwise explanation of the procedure of the HGOAAM fitting.

Algorithm 4.3.1: MAIN LOOP()

$[t]P[t]$ population of size N is created randomly at t=0
while stopping criterion is not met
do
- $P[t] \leftarrow Sort(P[t])$
- $Q[t, 0 : N/2] \leftarrow Selection(P[t])$
- $Q[t] \leftarrow Reproduction(Q[t])$
- **if** $t > b$
  - **then** $\{Jacobian \leftarrow Jacobian \cup GradientOperator(Q[t])$
- **if** $(P_{J_{param}} \geq 1)$
  - **then** $\begin{cases} Q[t, N/2 : N] \leftarrow GradientDescent(P[t, 0 : N/2]) \\ clearJacobian \end{cases}$
  - **else** $\{Q[t, N/2 : N] \leftarrow P[t, 0 : N/2]$
- $P[t+1] \leftarrow Q[t]$
- $t \leftarrow t + 1$

$Solution \leftarrow Min(P[t])$

---

1. **Initialization:** Test image is loaded, along with the location of center of gravity (COG) of the unknown face which can be estimated by a face detector. Initialization of a population P[t] of N chromosomes is carried out comprising of appearance parameters C and pose parameters P.

2. **Segmentation:** Each chromosome corresponds to a 3D shape. Each shape is deformed, rotated and translated according to appearance, rotational and translational parameters in the chromosome. This deformed shape is placed on the test image to warp the face to mean frontal shape as shown in the figure 3.3. Photometric texture normalization is applied on the warped image to overcome illumination variations.

3. **Fitness:** Pixel error (fitness) is then calculated by equation 4.1 between this warped image and frontal view image of the database obtained by the appearance parameters of each chromosome.

4. **Reproduction:** After calculating the fitness of each chromosome, tournament selection is performed for the reproduction of next generation. Half of this population is selected through tournament selection and their chromosomes are crossed over and mutated with the probability rate of P_x and P_m respectively. The parents are replaced with the newly born children to make half of a new population for segmentation phase. The second half of the population is copied directly after sorting with respect to pixel error.

5. **Gradient Operator:** During mutation of each gene in a chromosome, error image is stored for calculating Jacobian matrices. As explained in previous section, after b number of generations, whenever mutation occurs partial difference of the error function is calculated with respect to each gene (parameter). Finally after further evolutions of generations, when all the genes undergoes mutation, arithmetic mean of Jacobian matrices is calculated to apply on the half of the population of the current generation. This half of the population was being copied directly in the step of reproduction of previous generations (as explained in previous step) and now it is updated by the gradient operator.

Steps 2 to 5 are repeated until stopping criterion of particular number of generations is fulfilled while saving the best fit chromosome. Chromosome of the best result min(P[t]) contains the best appearance and pose parameters for a given face.

4.4 Experiments and Results

We performed simulations using 2.5D AAM model made on publicly available database of M2VTS. 2.5D AAM model, of the size of 64 by 64 pixels, is created by annotating profile view and frontal view images of all 37 subjects of M2VTS database. 2.5D AAM model is acquired along with its C and P parameters. Parameter C is constrained by $\pm 2\sqrt{\lambda}$, where λ are the eigenvalues obtained by applying PCA and retaining 95% of the variation in equation 3.4. Whereas pose parameters P varies as shown in table 4.2.

Pose Params	**Min. Value**	**Max. Value**
θ_{yaw}	$-60°$	$60°$
θ_{pitch}	$-5°$	$5°$
θ_{roll}	$-5°$	$5°$
t_x	-10% of MS	10% of MS
t_y	-5% of MS	5% of MS
$Scale$	-5% of MS	5% of MS

Table 4.2: Pose Parameters (MS = Model Size)

In testing phase we apply this model on totally different face databases of SUPELEC'08 (contains 246 images of 7 individuals), Pointing'04 (contains 390 images of 15 individuals) and synthetic (contains 600 images of 5 individuals) facial images.

Three sets of experiments of optimizations are performed. i) HGOAAM (Hybrid Genetic Optimization for AAM) ii) Gradient Descent (GD) and iii) HGA-Sim (Hybrid GA-Simplex). In the experiment of HGOAAM and HGA-Sim size of the population is 140 chromosomes and their 25 generations are evolved in each facial image analysis as tabulated in table 4.3. The size of the population and the number of generations are chosen by performing various simulations, while taking into account available time and memory resources. The type of encoding used in these experiments is Value Encoding. Different experiments are performed by applying uniform, single point, two point and arithmetic crossover on the chromosomes. From these experiments the two point crossover is chosen as others do not make significant difference in the results. Similarly among random and Gaussian mutation, Gaussian mutation is chosen which works with respect to the distribution formed by the associated limits of the C and P parameters.

In the experiments by GD, different configurations are tested and only the best is taken, in which Jacobian matrices with respect to each parameters is calculated and model is applied on a grid of 27 points of initializations. These 27 initializations are in the forms of an 3x3x3 array, where each dimension corresponds to t_x, t_y and $Scale$ of the model.

Population	140
Generations	25
b	10
Cross-Over	Two-point
P_x	80% (of population)
Mutation	Gaussian
P_m	2% (genewise probability)
P_k	20% (of population)
Selection	Tournament

Table 4.3: Specifications of Genetic Algorithm's parameters

Best chromosomes obtained at the end of the experiments, contain the localization of features like eyes, nose and mouth. Figures 4.6, 4.7 and 4.8 show the comparison of applying above mentioned three algorithms on face images of three facial image databases. It can be seen from the facial images that facial features are better localized and pose is well estimated in HGOAAM (figures 4.6(a), 4.7(a) and 4.8(a)) than the other two algorithms.

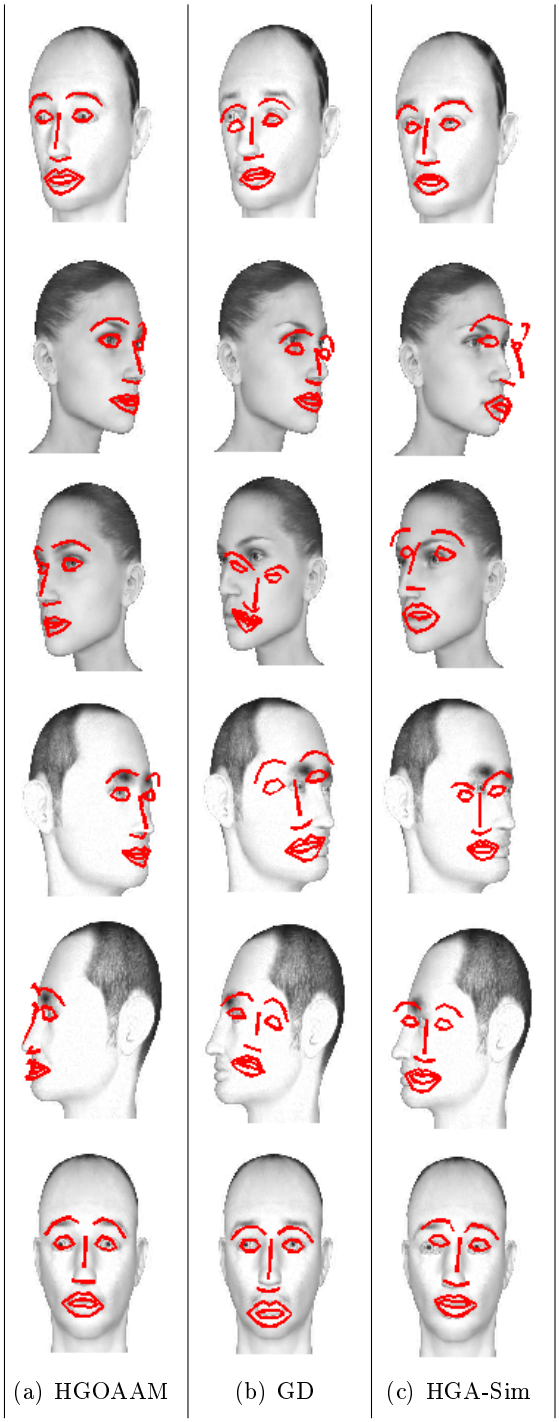
After obtaining the best solution, ground truth error is calculated between the facial features localized by the solution and the ground truth points marked manually as explained in the section 3.3 of chapter 3. For the quantitative analysis figures 4.9, 4.10 and 4.11 show comparison of above mentioned three methods by plotting percentage of aligned images of each database versus ground truth error. In these figures we compare the algorithms within ground truth error of 15% of D_{eye}. It can be seen from the images that HGOAAM outperforms the other two methods, for instance in figure 4.9 within an error of 15% of D_{eye} (distance between eyes), GD was able to align 20%, HGA-Sim

(a) HGOAAM (b) GD (c) HGA-Sim

Figure 4.6: Localization of facial features of Pointing'04 facial images by HGOAAM, GD and HGA-Sim

(a) HGOAAM (b) GD (c) HGA-Sim

Figure 4.7: Localization of facial features of SUPELEC'08 facial images by HGOAAM, GD and HGA-Sim



(a) HGOAAM (b) GD (c) HGA-Sim

Figure 4.8: Localization of facial features of synthetic facial images by HGOAAM, GD and HGA-Sim

was able to align 30%, whereas HGOAAM aligned 41% of the POINTING'04 facial images. Similarly figure 4.10 illustrates that GD aligned 47%, HGA-Sim aligned 46% and HGOAAM aligned 58% of the facial images of SUPELEC'08 database. For synthetic facial images the figure 4.11 illustrates that GD aligned 35%, HGA-Sim aligned 36% and HGOAAM aligned 46% of the images within ground truth error of 15%. While comparing the results of SUPELEC'08 facial images with synthetic facial images, we found that facial features of SUPELEC'08 faces are localized more accurately than that of synthetic faces. It is due to the difference in the texture of the skin, which is uniform in the case of synthetic faces while SUPELEC'08 faces have more natural skin similar to the learning database's faces of M2VTS. As far as the results of POINTING'04 facial images are concerned, their illumination conditions are not similar to the other two databases, which makes their results less accurate.

For the accuracy of pose estimation we have compared the results of only synthetic facial due to availability of accurate value of θ_{yaw}. Figure 4.12 shows that within the error of $\pm 5°$ of θ_{yaw}, HGOAAM was able to estimate pose of 26% of total images compared to GD-21% and HGA-Sim-13%. Table 4.4 gives detailed analysis and comparison of these algorithms in terms of accuracy, efficiency and robustness.

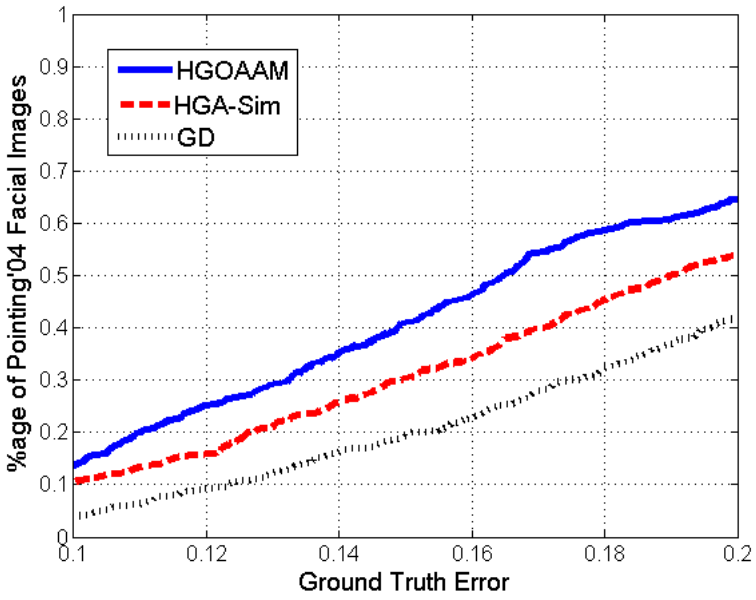


Figure 4.9: Ground truth error comparison of HGOAAM, GD and HGA-Sim for Pointing'04 Database.

As far as time consumption is concerned, GD required approximately 185 warps for each initialization, that makes 4995 warps with three initializations of each t_x, t_y and *Scale*. HGOAAM and HGA-Sim required 2500 warps (140 warps of initial population, 70 warps for 25 generations evolution and 610 warps in GD or Simplex) to localize even a profile view face from scratch. Each warp equals 90% of the total time consumed by

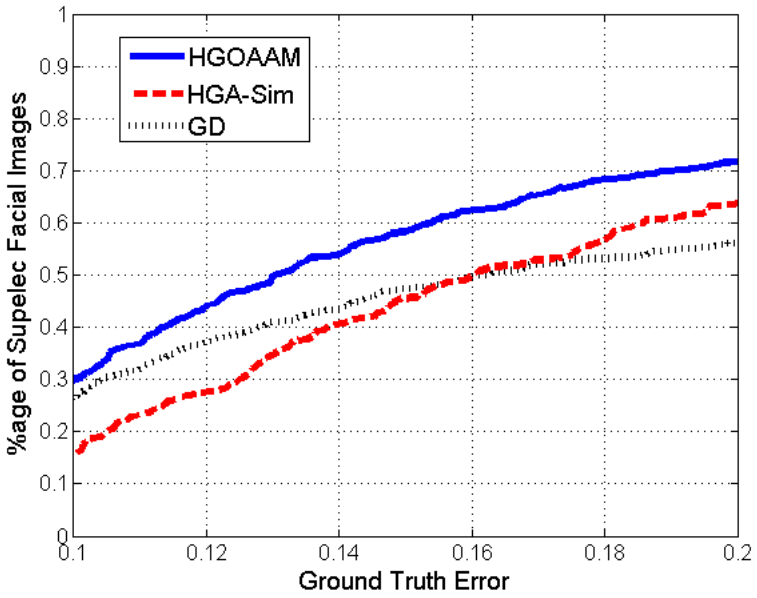


Figure 4.10: Ground truth error comparison of HGOAAM, GD and HGA-Sim for SUPELEC'08 Database.

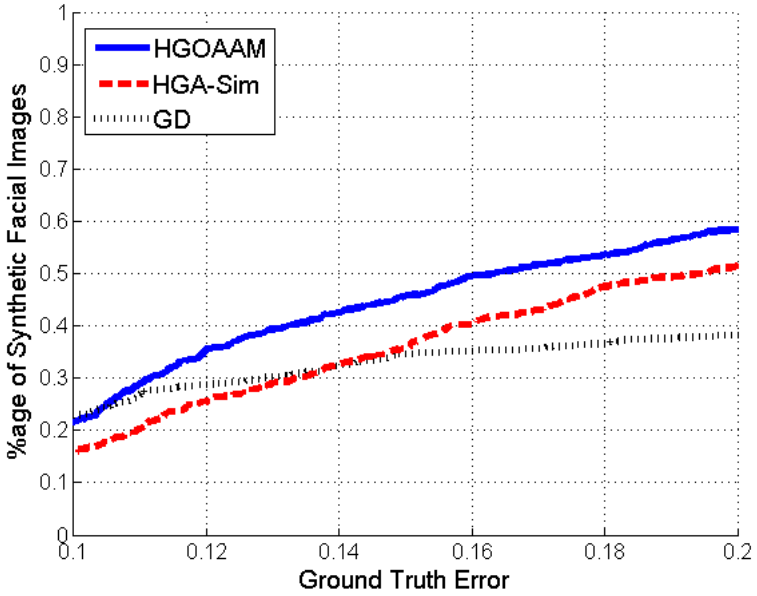


Figure 4.11: Ground truth error comparison of HGOAAM, GD and HGA-Sim for Synthetic Database.

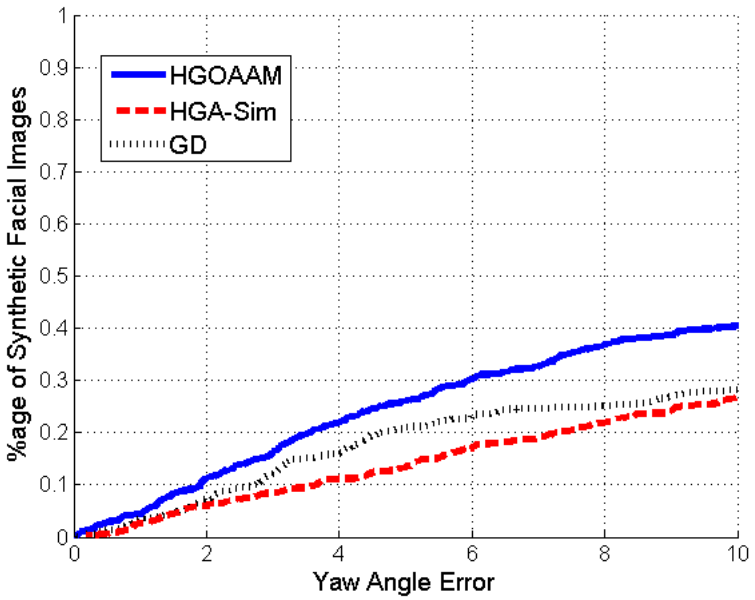


Figure 4.12: Comparison of θ_{yaw} error between HGOAAM, GD and HGA-Sim for Synthetic Database

Method	**Face**	**Images Aligned** ($GTE \leq 15\%$)	**Pose Estimated** ($E_{(\theta_{yaw})} \leq 5°$)	**No. of Warps**	**Time (msec)**
HGOAAM	POINTING'04 (390)	**41%**	—	**2500**	**83**
GD		20%	—	4995	166
HGA-Sim		30%	—	**2500**	**83**
HGOAAM	SUPELEC'08 (246)	**58%**	—	**2500**	**83**
GD		47%	—	4995	166
HGA-Sim		46%	—	**2500**	**83**
HGOAAM	Synthetic (600)	**46%**	**26%**	**2500**	**83**
GD		35%	21%	4995	166
HGA-Sim		36%	13%	**2500**	**83**

Table 4.4: Analysis of the Results

an iteration i.e. 0.03 msec in Pentium-IV 3.2GHz. An overhead of calculating Jacobian matrices in the case of HGOAAM do not require that much amount of time than the accuracy achieved by it. Thus for a complete facial analysis of a face HGOAAM requires 83 msec in Pentium-IV 3.2GHz, which means it can successfully analyze 12 frames in one second.

4.5 Conclusions

The solution proposed in this chapter is to extract the facial features of an unknown face making large lateral movements in front of a single camera with out any prior knowledge of the face's pose and appearance. We have proposed an efficient optimization technique by the hybridization of genetic algorithm (GA) with gradient descent (GD) to make a robust, efficient and real time facial search optimization for 2.5D AAM. Simple hybridization of gradient descent in GA makes the system computationally expensive. Therefore for this hybrid optimization we propose a gradient operator in GA, which functions (calculates gradients of the solutions) in conjunction with the existing genetic operator of mutation. Thus it does not increase the computational cost of the system and achieve the said efficiency and robustness without introducing complex computations. Our algorithm has been tested on facial images from Pointing'04, SUPELEC'08 and synthetic databases to extract the facial features of the faces making large lateral movements. Results of the comparison of our proposed algorithm of HGOAAM with optimization techniques of classical gradient descent (GD) and a hybrid GA-Simplex have shown that our proposition outperformed both the algorithm in terms of accuracy, robustness and efficiency.

In the next chapter we will make use of the contribution of this chapter and modify it to be used in multiple camera configuration. Although multiple camera help in analysing oriented faces more effectively than single camera due to multiple informations, but for that it requires multi-objective optimization for the face search. All of these issues will be discussed in the next chapter.

Chapter 5

AAM fitting for Multiple View Images

Contents

This chapter presents the third contribution of the thesis. It describes the facial analysis of the multi-view or multi-camera images by 2.5D AAM. Searching for an optimum solution of two or more distinct information from multiple cameras requires multi-objective optimization (MOO). First section 5.1 introduces the MOO and presents the related work around some of the well known Pareto-based MOO techniques. Section 5.2 introduces the concept of MOO for the face search by AAM in a multi-view scenario. Section 5.3 discusses a new concept of hybrid multi-objective optimization and how it can be implemented in a multi-view face analysis system by 2.5D AAM. In this section hybridization of multi-objective optimization algorithms in different domains are elaborated by referring to the articles of various authors, followed by the explanation of a unique way of hybridization of genetic algorithm and gradient descent methods to make HMOAAM (Hybrid Multi-objective Optimization for AAM) more robust and efficient. Section 5.3.3 explains the stepwise application of our algorithms of HMOAAM on the facial images. While section 5.4 presents the experiments and results of applying HMOAAM, MOAAM (Multi-objective Optimization for AAM) and SOAAM (Single-objective Optimization for AAM) on facial images databases.

In single-view or single camera system, single error between model and query image is optimized. However in multiple camera system, optimization of more than one error is to be performed between a model and test images from each camera. The objective is to minimize all the pixel errors $e_1, e_2, ..., e_M$ of equation 3.6 obtained by M cameras

$$e_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[I_{i,j}(C, P_j) - M_i(C)]^2} \tag{5.1}$$

where j varies from 1 to M and $M \geq 2$. P_j are the pose parameters and are linked by offsets of rotational and translational parameters obtained in calibration (section 3.2.2.2 of chapter 3). N is the number of pixels of the model. In multi-view AAM, model is overlaid on the images from each camera simultaneously. In order to optimize all the pixel errors simultaneously, multi-objective optimization for the search by 2.5D AAM is proposed.

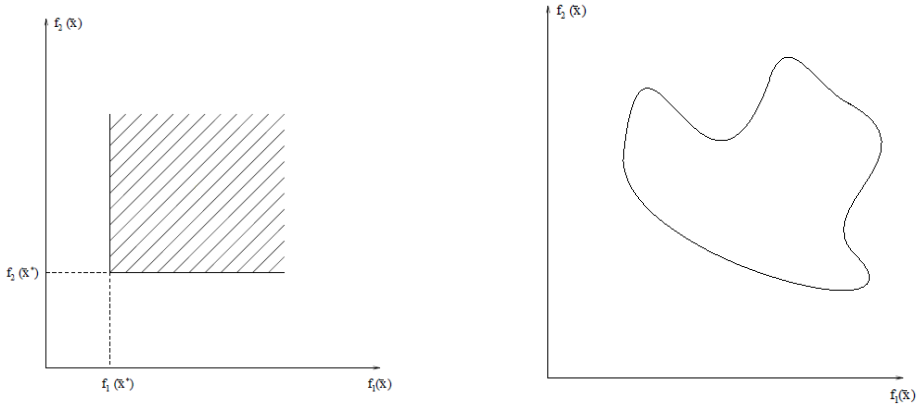
5.1 Multi-Objective Optimization

Multi-objective optimization (also called multi-criteria optimization, multi-performance, or vector optimization) can be defined as the problem of finding

"a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the designer." [108].

Formally, we can state this as, "find the vector $x = [x_1, x_2, ...x_n]^T$ that will optimizes the vector function $f(x) = [f_1(x), f_2(x), ..., f_k(x)]^T$". In other words, we wish to

determine the particular set $x = [x_1, x_2, ...x_n]^T$ that yields the optimum values of all the objective functions. The problem is that the meaning of optimum is not well defined in this context. In this case x would be a desirable solution, but normally we never have a situation in which all the $f_i(x)$ have a minimum at a common point. An example of the ideal situation is shown in 5.1(a). However, since this situation is rare in real-world problems as illustrated in figure 5.1(b), we have to establish a criterion to determine what is an *optimal* solution with respect to all the objective functions.



(a) Ideal solution in which all the functions have their minimum at a common point.

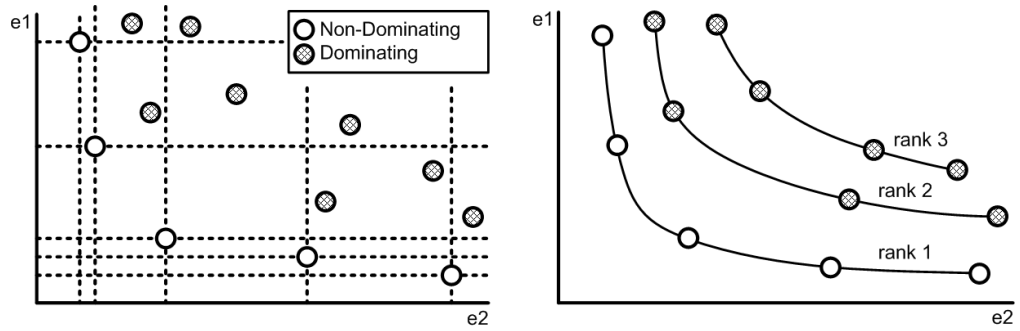
(b) Real world problem representation.

Figure 5.1: Bi-objective problem representation

5.1.1 Pareto Optimum and Pareto Front

The concept of a *Pareto optimum* was formulated by Vilfredo Pareto in the nineteenth century [109], and by itself constitutes the origin of research in multi-objective optimization. The definition says that in a minimization context x is Pareto optimal if there exists no feasible vector that decreases some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, the Pareto optimum never gives a single solution, but rather a set of solutions called *non-inferior* or *non-dominated* solutions.

The minima in the Pareto sense are going to be in the boundary of the design region of the objective functions. In figure 5.2, line drawn on dots shows this boundary for a bi-objective problem. In general, it is not easy to find an analytical expression of the line or surface that contains these points, and the normal procedure is to compute the points and their corresponding solutions. When we have a sufficient number of these, we may proceed to find the Pareto non-dominated solutions by the rest of the solutions. These solutions are assigned the highest rank and are removed from further assignment of the ranks. Remaining population undergoes the same process of ranking until the solutions are suitably ranked in the form of *Pareto fronts*.



(a) (b)

Figure 5.2: Pareto Fronts

5.1.2 Pareto-Based MOO Approaches

The idea of using Pareto-based fitness assignment was first proposed by Goldberg [90] for Genetic Algorithms (GA). He suggested the use of non-domination ranking and selection to move a population of chromosomes toward the Pareto front in a multi-objective optimization problem. The basic idea is to find the set of chromosomes in the population that are Pareto non-dominated by the rest of the population. These chromosomes are then assigned the highest rank of rank 1 (first Pareto front is illustrated in figure 5.2(b) by small empty circles) and eliminated from further contention. Another set of Pareto non-dominated strings are determined from the remaining population and are assigned the next highest rank of rank 2, rank 3 and so on (second and third Pareto fronts are illustrated in figure 5.2(b) by small filled circles). This process continues until the population is suitably ranked. Goldberg [90] also suggested the use of some kind of diversity to keep the population from converging to a single point on the front. Some of the multi-objective optimization techniques based on the genetic algorithms are explained below.

5.1.2.1 MOGA

Fonseca and Fleming [110] proposed a scheme in which the rank of an individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual x_i of generation t dominated by $p_i^{(t)}$ individuals in the current generation. Its current position in the population can be given by [110]:

$$rank(x_i, t) = 1 + p_i^{(t)}$$

All non-dominated individuals are assigned rank 1, while dominated individuals are penalized according to the population density of the corresponding region of the trade-off surface. Fitness assignment is performed in the following way:

(1) Sort population according to rank.

(2) Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $n \leq N$) as proposed by Goldberg [90], according to linear function.

(3) Average the fitnesses of individuals with the same rank, so that all of them are selected with the same probability. This procedure keeps the global population fitness constant while maintaining appropriate selective pressure.

This type of blocked fitness assignment is likely to produce a large selection pressure that might produce premature convergence. To avoid this, they use a niche-formation method to distribute the population over the Pareto-optimal region, but instead of performing sharing on the parameter values, they use sharing on the objective function values. In MOGA, sharing is done on the objective value space, which means that two different vectors with the same objective function values can not exist simultaneously in the population. Moreover its performance is highly dependent on an appropriate selection of sharing factor, thus needs to develop a good methodology for its computation.

5.1.2.2 NPGA

Horn et al. [111] proposed a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals, a number of individuals is used to help determine dominance (t_{dom}). When both competitors are either dominated or non-dominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing. Population sizes considerably larger than usual with other approaches are used, so that the noise of the selection method is tolerated by the emerging niches in the population.

Since this approach does not apply Pareto selection to the entire population, but only to a segment, the technique is very fast and produces good non-dominated solutions that can be kept for a large number of generations. However, to perform well, besides requiring a sharing factor, this approach also requires a good choice of the value t_{dom}, complicating its appropriate use in practice. Moreover population sizes larger than usual also increases its complexity.

5.1.2.3 NSGA

The non-dominated sorting genetic algorithm (NSGA) was proposed by Srinivas and Deb [112]. Before selection, the population is ranked on the basis of non-domination: All non-dominated individuals are classified into one category (with a dummy fitness value, which is proportional to population size to provide equal reproductive potential for these individuals). To maintain diversity in the population, classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of non-dominated individuals is processed. The process continues until all individuals in the population are classified. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows the search for non-dominated regions and results in quick convergence of the population toward such regions. Sharing, for its part, helps to distribute the population over this region. NSGA efficiency lies in the way in which

multiple objectives are reduced to a dummy fitness function using a non-dominated sorting procedure. With this approach, any number of objectives can be solved, and both maximization and minimization problems can be handled.

In this case, sharing is done in the parameter values instead of the objective values, to ensure a better distribution of individuals, and to let multiple equivalent solutions exist. However, this technique is more inefficient (both computationally and in the quality of Pareto fronts produced) than MOGA, and more sensitive to the value of the sharing factor.

5.1.2.4 NSGA-II

Multi-objective evolutionary algorithms which use non-dominated sorting and sharing have been mainly criticized for their computational complexity, non-elitism approach and the need for specifying a sharing parameter. NSGA-II proposed by Deb et al. [6] alleviates all these difficulties.

Initially, a random parent population P[t] is created at t=0. Fitness of each solution is calculated and they are sorted based on the non-domination sort (A). Each solution is assigned a rank equal to its non-domination level (1 is the best level). Tournament selection, crossover, and mutation operators are used to create a child population Q[t] of size N. Secondly, a combined population $R[t] = P[t] \cup Q[t]$ is formed. The population R[t] will be of size 2N. Then, the population R[t] is sorted according to non-domination. The new parent population P[t+1] is formed by adding solutions from the first front till the size exceeds N. Thereafter, the solutions of the last accepted front are sorted according to the crowding distances and the first N points are picked. This population of size N is now used for selection, crossover and mutation to create a new population Q[t+1] of size N. This process is repeated for a particular number of generations until the stopping criteria of the algorithm is met. Detailed description of NSGA-II algorithm along with its pseudocode is given in appendix A.

From the above discussion NSGA-II proves to be one of the best Pareto based multi-objective optimization technique compared to other known techniques. In this thesis NSGA-II is implemented for the face search by hybrid multi-objective AAM.

5.2 Multi-Objective AAM (MOAAM)

This section explains the multi-objective optimization for face search by AAM. As discussed earlier, in multi-camera system, optimization of more than one error is to be performed. Therefore the objective is to minimize all the pixel errors $e_1, e_2, ..., e_M$ of equation 5.1 obtained from M cameras.

AAM fitting on multi-views from M cameras is shown in figure 5.3. In this multi-view AAM, model is overlaid on all the images from each camera with the same C parameters. P parameters of all the AAM models are linked with the offsets $P_{offset,j}$. In order to optimize M pixel errors simultaneously, Pareto based optimization by NSGA-II is proposed.

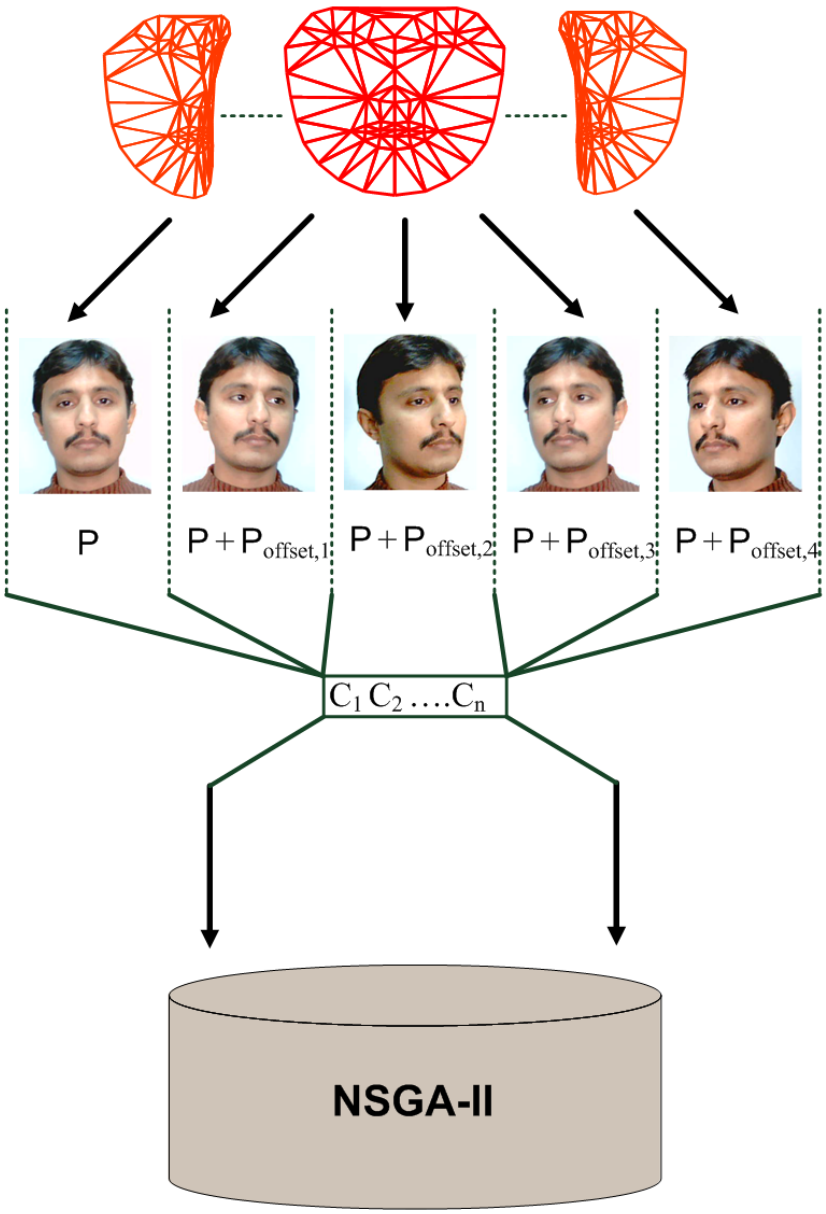


Figure 5.3: Fitting of AAM on multiple views

MOAAM uses NSGA-II proposed by Deb et al. [6] to optimize the appearance C and pose parameters P. The target is to find out the best possible values of these parameters giving minimum pixel errors between the model and the query images of all the cameras. In this optimization technique each parameter is considered as a gene. All the genes of C and P are concatenated to form a chromosome as shown in the figure 5.4. Population of particular number of chromosomes is randomly created. Pixel errors (fitness) between query images and model represented by each chromosome are calculated. Tournament selection is applied to select parents from the population to undergo reproduction. Two point crossover and Gaussian mutation is implemented to reproduce next generation of the chromosomes as shown in the figure 5.5. Selection and reproduction is based upon non-dominating sort. The objective is to minimize M pixel errors, hence non-dominating scenario is used as explained in section 5.1.2.4.

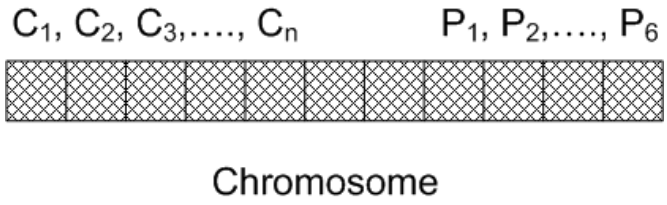


Figure 5.4: Chromosome

Using direct search methods like NSGA-II has some drawbacks. Their speed of convergence do not let the system to remain real time. Although stopping criteria can be varied to make them real time, but in that case their robustness is reduced. Therefore this thesis concentrate on hybridizing the MOAAM with gradient descent method to achieve high robustness in real time.

5.3 Hybrid Multi-Objective Optimization

Over the last decade hybrid evolutionary algorithms, also known as memetic algorithms, has gained a lot of interest of researchers [113]. This area of research has been tremendously increasing to solve the real world optimization problems. Researchers have hybridized different kinds of optimization techniques. The most common combination is of evolutionary algorithms with deterministic methods. Their successful hybridization in single-objective optimization problems has shown promising results. However in multi-objective optimization, studies are limited, especially in the domain of facial analysis.

In face analysis hybridization is required due to the non-convexity of the multidimensional facial search space formed in face alignment by AAM. As long as face stays in front of the camera, the error between AAM model and facial image remains convex. The moment it moves laterally various local optima appears in its error curve as shown in the figure 4.1 of chapter 4. This non-convexity requires search methodologies which could exploit and explore the search space at the same time. As discussed in chapter

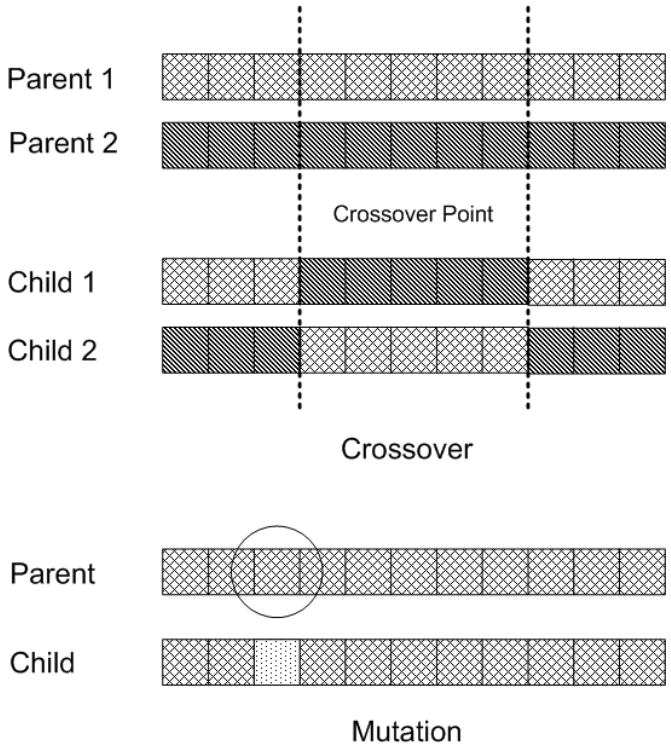


Figure 5.5: Reproduction Operators

4, exploitation property of gradient descent and exploration property of GA could be hybridized to make an efficient optimization system.

5.3.1 Previous Work

The following section presents the work of some authors who have improved the efficiency of their system by using hybrid multi-objective algorithms in their respective domains.

Lahanas et al. [114] hybridized MOEA (NSGA-II) with deterministic gradient-based methods for the dose optimization problem in high-dose rate brachytherapy or intensity modulated radiotherapy. Li et al. [115] hybridized GA and simulated annealing (SA) along with adaptive crossover and mutation operator for dosimetric optimization of external beam radiation. Wanner et al. [116] described hybrid MOGA employing local search procedure of quadratic approximation-based for optimization in electromagnetics. They use past samples of the previous generations to generate quadratic approximations to improve an individual locally.

Bosman and Jong [117] applied GD after each generation of MOEA and take only those solution which either improve in one objective or remain same. They tested the algorithm on a few well-known benchmark problems. Hiwa et al. [118] described a hybrid optimization using DIRECT (Dividing Rectangles), GA and SQP (Sequential Quadratic programming) for global exploration of bench mark problems e.g. Rosenbrock, Rastrigin and Schwefel functions. They also used simplex crossover operator in GA. Martinez and Coello [119] combined NSGA-II with Nelder and Mead's simplex method to compare the robustness with and with out hybridization by scalable test problems for evolutionary algorithm. Yildiz et al. [120] presented a two stage optimization technique. In the first stage, they implement Taguchi's robust parameter design to define robust initial population levels of design parameters to achieve better initialization in the second stage of genetic algorithm search.

No doubt all the experiments by these hybrid evolutionary algorithms showed an improvement over the conventional optimization, but the proposed hybridizations are time consuming and can not be implemented in real time. Moreover most of the researchers have proven the robustness of hybrid algorithms on well known synthetic mathematical problems rather than highly complex real world problems. On the other hand the hybridization technique discussed in chapter 4 for single objective optimization is less time consuming and can be modified for multi-objective optimization problems. In the next section we provide detailed description of the hybridization of GD with NSGA-II for the real time face search optimization in AAM.

5.3.2 HMOO in AAM

In this section we present a facial analysis system based on facial images captured by multiple cameras and analyzed by 2.5D AAM optimized by hybridization of NSGA-II and gradient descent. Calculations of gradient in GD makes the system time inefficient, therefore section 5.3.2.1 proposes an efficient procedure to calculate the gradients of a

function during the evolution of generations in NSGA-II. It explains how GD uses the mutation operator of NSGA-II for its pre-computations. Section 5.3.2.2 describes the hybridization by an efficient procedure of extracting meaningful solutions from the population formed by NSGA-II and transfer to the subsequent phase of GD optimization.

5.3.2.1 Gradient Operator

Here we make use of our previous proposition of gradient operator discussed in section 4.3.1. This operator is modified for the multi-objective optimization by NSGA-II. Simple integration of gradient descent with NSGA-II would have increased the number of error evaluations, but this operator functions in conjunction with mutation operator. Thus gradient operator uses the error evaluation of mutation operator and do not put an extra burden on the system.

As discussed in section 4.3.1, during mutation of a chromosome only one gene or parameter is changed for next generation. Error evaluation of this variation is retrieved by gradient operator as the partial differential of all the functions. Thus $J_{j,C}$ and $J_{j,P}$ represents the mean partial differential of M error functions from each camera with respect to each parameter. The remaining procedure of calculating ΔC and ΔP is applied in the second phase of the optimization with the help of the factor called CIRF.

5.3.2.2 Camera Information Relevance Factor (CIRF)

As mentioned earlier NSGA-II is a population based algorithm therefore it has some advantages and drawbacks. In multiple camera system when we use population based multi-objective optimization algorithm like NSGA-II we have an advantage of analyzing the population formed by the facial information from all the cameras at the same time.

Let us consider two webcams $M = 2$ installed at the extreme edges of the display, the face can be made visible in the three possible regions of R1, R2 and R3 as shown in the figure 5.6, which is a modified version of figure 3.11. If the face is visible in region R1 both the cameras have relevant facial information therefore the formation of chromosomes is uniform with respect to each pixel error. On the contrary if the face is visible in region R2 and R3 then the chromosome formation is inclined towards one of the pixel error. Synthetic representation of the fronts formed by chromosomes are shown in figure 5.6, whereas figures 5.7(a), 5.7(b) and 5.7(c) are the actual representation of chromosomes formation. This arrangement of the chromosomes for each orientation of face compelled us to calculate a factor called *Camera Information Relevance Factor (CIRF)*, which expresses the relation of facial image information from each camera and ultimately will be used for the subsequent Gradient Descent optimization phase.

Let us consider that at any instance face is oriented in such a way that facial information from both the cameras is valid but the rate of their relevant information varies. To determine this ratio we develop a technique to analyze the NSGA-II population optimally ranked by Pareto. The procedure is started after meeting the stopping criteria of the optimization phase of NSGA-II for a given facial image. It consists of i) calculating the medians of both cameras pixel errors $e_{1,i}$ and $e_{2,i}$ of N chromosomes of the popula-

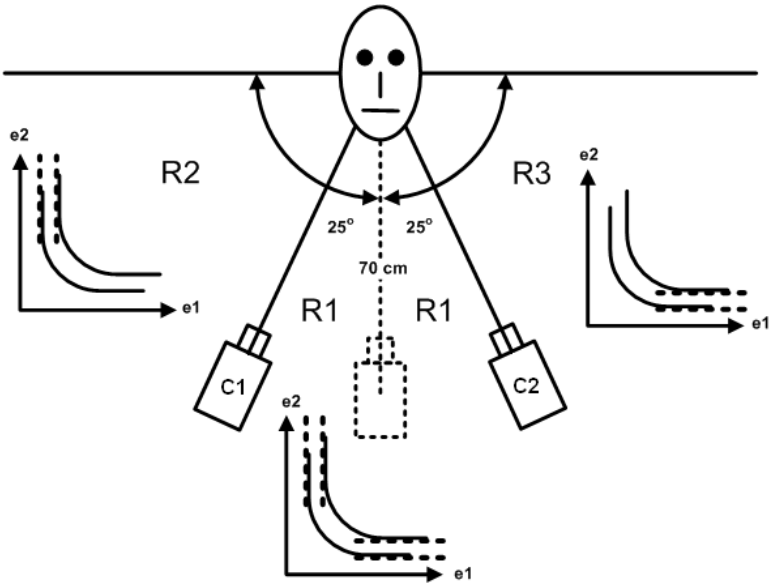


Figure 5.6: Multi-View System

tion, ii) extend a line between this median value and the minimum among the minima of both the errors of the current population ($e_{1,i}$ and $e_{2,i}$) as shown in the figures 5.7(a), 5.7(b) and 5.7(c). Using these values CIRF can be calculated mathematically for each facial orientation as

$$\psi = \frac{arctan\left(\frac{\tilde{e_2}-e_{min}}{\tilde{e_1}-e_{min}}\right)}{\pi/2} \tag{5.2}$$

where

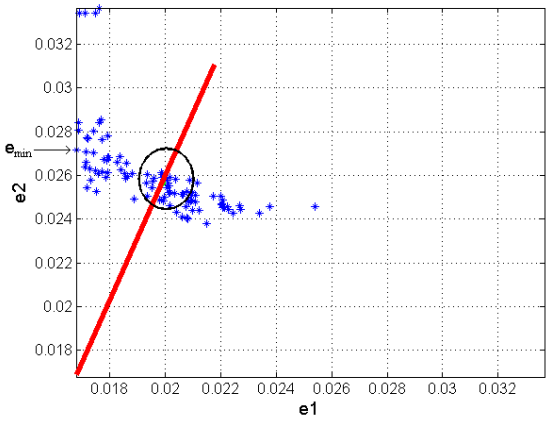
$$\psi = Camera\ Information\ Relevance\ Factor,$$

$$\tilde{e_1} = Median\ of\ e_{1,i} \quad 0 \leq i \leq N,$$

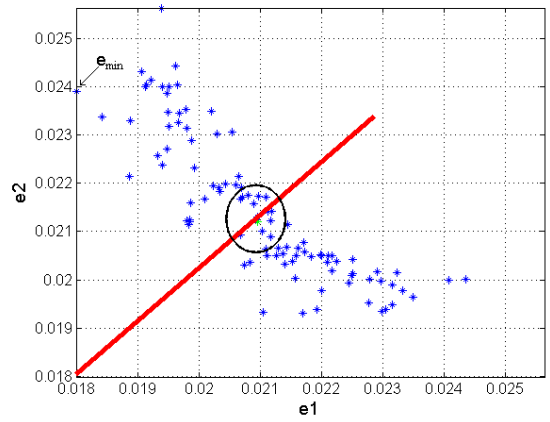
$$\tilde{e_2} = Median\ of\ e_{2,i} \quad 0 \leq i \leq N,$$

$$e_{min} = Minimum\ (Minimum(e_{1,i})\ ,\ Minimum(e_{2,i}))$$

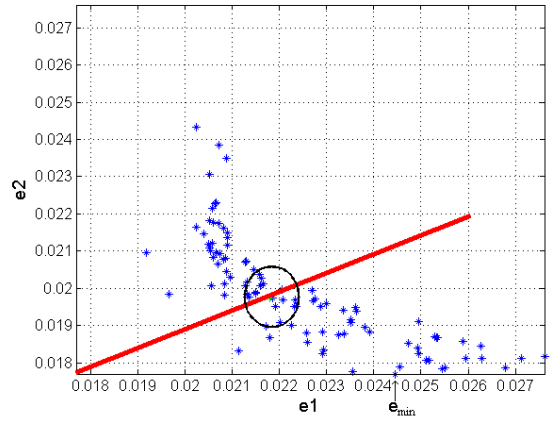
Figure 5.7(a), 5.7(b) and 5.7(c) represent CIRF equal to 0.79, 0.52 and 0.30 respectively. This means that dominant cameras of these figures respectively are camera1, both cameras and camera2. Value of CIRF varies from zero to one. With respect to one of the camera, if a face is oriented such a way that it occludes itself then the value of CIRF should become one or zero. However due to the diversity of NSGA-II, it never become one or zero although it approaches to two extremes. This extremely small or large value would automatically eliminate the information of the camera unable to provide information of the face due to its orientation. Figure 5.8 illustrates the values of CIRF when a face is rotated laterally from $-60°$ to $+60°$ in front of our system of double camera. One can see how the CIRF gradually increases and decreases from 0.5



(a) Population formation when the face is in front of Camera1



(b) Population formation when the face is equally visible from Camera1 and Camera2



(c) Population formation when the face is in front of Camera2

Figure 5.7: Population formation when face moves laterally from $-60°$ to $+60°$

when the face rotates towards each camera. This value of CIRF is used as a weighting coefficient in subsequent gradient descent optimization phase as follows

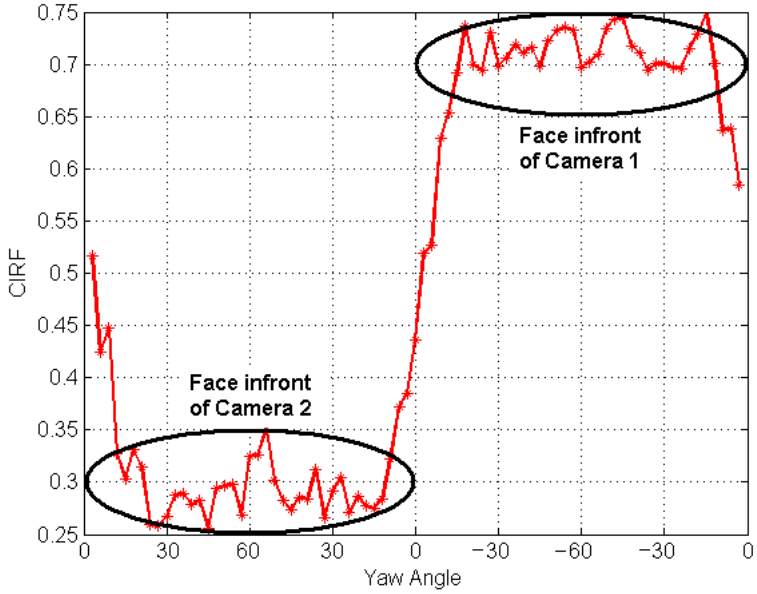


Figure 5.8: Variations of CIRF for a face moving laterally from $-60°$ to $+60°$

$$e_{total} = e_1 * \psi + e_2 * (1 - \psi) \tag{5.3}$$

With this equation we are able to automatically choose the most relevant information from multiple cameras and further proceed for the analysis.

Now the matter is to select the set of chromosomes, from the NSGA-II population, for the initialization of GD. The basic theme behind the evolutionary algorithms, that the fittest will prevail, remains the same whether it is multi-objective or single objective optimization. Therefore chromosomes behavior is understood that they will try to reproduce the most fittest chromosomes of all and consequently can arrange themselves in different ways with respect to pixel errors between the model and the images from both of the cameras as shown in the figure 5.7. Apart from calculating CIRF, set of best fit chromosomes are also chosen with the help of the line drawn for calculating CIRF. All the chromosomes' pixel errors are sorted with respect to their shortest perpendicular distance from the line in order to maintain their own values of CIRF as close as possible to the one calculated by $\tilde{e_1}$ and $\tilde{e_2}$. Particular number of such chromosomes are selected from them (encircled in the figures 5.7(a), 5.7(b) and 5.7(c)), which are then used for the initialization of the upcoming calculations by GD.

5.3.3 HMOAAM fitting

This section describes pseudo code and the stepwise procedure of 2.5D AAM fitting by hybrid multi-objective optimization.

Algorithm 5.3.1: MAIN LOOP()

$P[t]$ population of size N is created randomly at t=0
$NondominatingSort(P[t])$
$Ranking(P[t])$
while stopping criterion is not met
do
$Q[t] \leftarrow Reproduction(P[t])$
$Jacobian \leftarrow Jacobian \cup GradientOperator(Q[t])$
$R[t] \leftarrow P[t] \cup Q[t]$
$NondominatingSort(R[t])$
$Ranking(R[t])$
$CrowdingDistance(R[t])$
$P[t+1] \leftarrow R[t, 0:N]$
$t \leftarrow t+1$
$(G[t], \psi) \leftarrow CalculateCIRF(P[t])$
$Solution \leftarrow GradientDescent(G[t], Jacobian, \psi)$

---

1. **Initialization:** Test images are loaded from M cameras, along with the location of center of gravity (COG) of the unknown face which can be estimated by a face detector. Uniformly distributed random initialization of a population $P[t]$ of N chromosomes is carried out comprising of appearance parameters C and pose parameters P. Fitness is calculated using equations 5.1 and the non-dominating sort of the entire population is carried out.

2. **Reproduction:** After initializing and calculating the fitness (pixel errors $e_1, e_2, ..., e_M$) of each chromosome, tournament selection is performed for the reproduction of next generation. Parents are selected, crossed over and mutated with the probability rate of P_x and P_m respectively to reproduce child population $Q[t]$ of N chromosomes. The parents are joined with the newly born children to make a bigger population of $2N$ chromosomes for non-dominated sorting.

3. **Segmentation:** Each chromosome corresponds to a 3D AAM shape. Each shape is deformed, rotated and translated according to appearance, rotational and translational parameters in a chromosome respectively. This deformed shape is placed on the test images of M cameras with an offset of $\theta_{(offset,yaw,j)}$, $\theta_{(offset,pitch,j)}$, $\theta_{(offset,roll,j)}$, $tx_{(offset,j)}$, $ty_{(offset,j)}$ and $Scale_{(offset,j)}$ of the j^{th} camera with respect to the central camera. Rest of the parameters i.e. C parameters remain same. The region covered by the deformed shapes is warped to mean frontal

shape as shown in the figure 3.3. Photometric texture normalization is applied on the warped image to overcome illumination variations. Offsets of rotational parameters are introduced into deformed shape models with respect to the central camera, which implies that model with out an offset represents a face viewed from central camera. We calculate the experimental results (as explained in section 3.3.1 of chapter 3) with respect to the facial image of this central camera.

4. **Fitness:** pixel errors $e_1, e_2, ..., e_M$ are then calculated (equations 5.1) between these warped frontal images and frontal view image obtained by the appearance parameters of each chromosome.

5. **Gradient Operator:** During mutation of each gene in the reproduction of chromosomes, error image is stored for calculating Jacobian matrices (section 5.3.2.1). Whenever mutation occurs partial difference of the error function is calculated with respect to each gene (parameter). Finally when every gene undergoes mutation, arithmetic mean of Jacobian matrices is calculated to apply in the second phase of optimization by GD.

6. **Non-Dominated Sorting:** Non-dominated sorting is performed on the entire population to form Pareto fronts and rank them as explained in section 5.1.1. As shown in figure 5.2 the two errors e_1 and e_2 of each chromosome are plotted and a rank is assigned to each chromosome. Similarly ranking can be performed for all the M errors. It is due to this ranking that we are able to sort chromosomes with respect to the ranks (in a multi-objective optimization) instead of pixel errors (in a single-objective optimization). Chromosome with equal ranks can be distinguished by the crowding distance of each chromosome.

7. **Replacement:** In reproduction step, size of the population becomes twice of the original size i.e. $2N$. To maintain constant size of the population only N chromosomes with lower Pareto fronts are selected for further generations. However if the chromosomes of the last rank are greater than N, then they are selected with respect to their crowding distances. Higher the crowding distance higher is the probability of selection for further generations.

Steps 2 to 7 are repeated until stopping criterion of particular number of generations is fulfilled while saving the best fit chromosome. The stopping criterion of fixed number of generations is just to make equal number of computations while comparing HMOAAM with other optimization techniques. Pareto fronts of this population are analyzed in order to calculate CIRF and to choose set of chromosomes for the initialization of GD as explained in section 5.3.2.2. Finally Jacobian matrices calculated during the generation evolutions, are applied on the chosen chromosomes to calculate the best solution by GD, which contains the best appearance and pose parameters for a given face.

We have presented the algorithm with M cameras. M can be equal to two or more than two, the principle remains the same if we use more than two camera. Number of Jacobian matrices will increase linearly with the increase in the number of cameras.

However Pareto fronts will be have M-dimensional representation, an example of triple objective Pareto fronts are shown in the figure 5.9. In the next section we apply the above algorithm for the optimization of face search in bi-objective or double camera system.

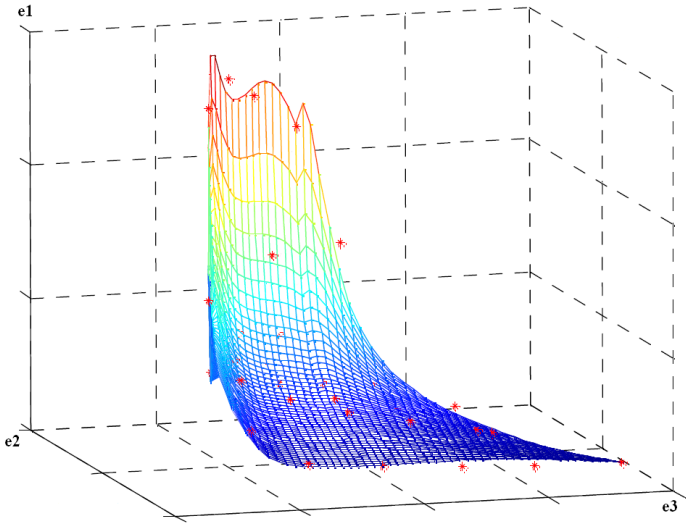


Figure 5.9: Pareto front in 3D

5.4 Experiments and Results

5.4.1 Experiments

In learning phase our proposed 2.5D AAM model (64 by 64 pixels) is constructed by annotating 37 subjects of publicly available databases of M2VTS [94]. 2.5D AAM model is acquired along with its C and P parameters (section 3.1 of chapter 3). Parameter C is constrained by $\pm 2\sqrt{\lambda}$, where λ are the eigenvalues obtained by applying PCA and retaining 95% of the variation in equation 3.4 on page 66. Whereas pose parameters variations is tabulated in table 5.1.

Whereas for test database we employed double camera SUPELEC database (246 images of 7 individuals) and double camera Synthetic database (4160 images of 52 individuals). In testing phase face alignment is performed on all the views from $-60°$ to $+60°$. Three sets of experiments are performed:

Single-Objective AAM: In SOAAM, 2.5D AAM is used for face alignment of facial images view from central camera. Classical GA is implemented for the optimization of face search.

Multi-Objective AAM: In MOAAM, NSGA-II alone is used for the facial analysis of the images from double camera system. The experimental details of the algorithm is given in the table 5.2.

Table 5.1: Limits of pose parameters (MS = Model Size)

Pose Parameters	Min. Value	Max. Value
θ_{yaw}	$-60°$	$60°$
θ_{pitch}	$-5°$	$5°$
θ_{roll}	$-5°$	$5°$
t_x	-10% of MS	10% of MS
t_y	-5% of MS	5% of MS
Scale	-5% of MS	5% of MS

Hybrid Multi-Objective AAM: In HMOAAM, NSGA-II hybridized with GD is used for the facial analysis of the images from double camera system (section 5.3.3). The experimental details of the algorithm is similar to MOAAM and given in the table 5.2 except the number of generations are reduced to 14 in HMOAAM.

Cameras (M)	2
Population	100
Generations	15
Cross-Over	Two-point
P_x	80% (of population)
Mutation	Gaussian
P_m	2% (genewise probability)
P_k	20% (of population)
Selection	Tournament Pool=4
Replacement	No
Diversity	Crowding Distance
$\theta_{(offset,yaw,1)}$	$25°$
$\theta_{(offset,yaw,2)}$	$-25°$

Table 5.2: Specifications of NSGA-II's parameters

As discussed in the previous chapter of single objective GA, the size of the population and the number of generations in HMOAAM and MOAAM are also carefully chosen by performing various simulations, while taking into account available time and memory resources. Encoding used in these experiments is also Value Encoding. Among uniform, single point, two point and arithmetic crossover, two point crossover is chosen as others do not make significant difference in the results. And the Gaussian mutation, which works with respect to the distribution formed by the associated limits of the C and P parameters, is chosen after comparing it with random mutation.

Stopping Criterion The stopping criterion of each algorithm is the fixed number of generations. Population size of chromosomes of each algorithm is 100 whereas their

number of generations or stopping criterion is different in order to allocate them equal computational time for the true comparison of their robustness. SOAAM, MOAAM and HMOAAM evolve respectively for 30, 15 and 14 generations. Selection, reproduction and replacement criteria are kept similar in all the above experiments.

Ground Truth Error As discussed in section 3.3.1, in double camera system, third central camera is placed in between the two cameras as shown in figure 5.6. Images from this camera is not used in the segmentation phase and only used to calculate GTE. AAM model obtained at the end of the facial analysis of an image from other two cameras is rotated and translated with respect to the coordinates of this third camera. Finally GTE of AAM model with respect to the facial image seen by this camera is calculated for the comparison of the methods.

5.4.2 Results

Best chromosomes, with respect to pixel errors, obtained at the end of HMOAAM, MOAAM and SOAAM contain best appearance and pose parameters for a given face. Features like eyes, nose and mouth can be extracted from the images with the help of the shapes given by chromosomes. This feature localization is shown in figure 5.10 for webcam facial images and figure 5.11 for synthetic facial images. It can be seen from these images that the feature localization gets far better in HMOAAM (figures 5.10(a) and 5.11(a)) than in MOAAM (figures 5.10(b) and 5.11(b)) and SOAAM (figures 5.10(c) and 5.11(c)).

Figure 5.12 shows percentage of aligned webcam images versus GTE of eyes, nose and mouth. Figure 5.12 depicts that our system of HMOAAM fitting is a lot better than MOAAM and SOAAM fitting. In HMOAAM 61% of the images are aligned with GTE less than 15% of the distance between the eyes, whereas MOAAM aligned 48% and SOAAM aligned 43% of the total images.

Similarly figure 5.13 shows percentage of aligned synthetic facial images versus GTE. It illustrates that HMOAAM aligned 48%, MOAAM aligned 39% and SOAAM aligned 22% of the synthetic facial images with GTE less than 15% of the distance between eyes. While comparing the results of real webcam images with synthetic facial images, we found that facial features of real faces are localized more accurately than of synthetic faces. It is due to the difference in the texture of the skin, which is uniform in the case of synthetic faces while real faces have more natural skin similar to the learning database's faces of M2VTS.

Figure 5.14 illustrates the percentage of synthetic facial images with respect to the exact estimation of the pose. This θ_{yaw} error estimation was possible only in synthetic facial image database due to the available ground truth value of θ_{yaw} (actual yaw angle) given by software MAYA. The figure 5.14 depicts that HMOAAM estimated the pose of 42% of synthetic facial images within the error of $\pm 5^\circ$ of θ_{yaw}, while MOAAM and SOAAM estimated 37% and 17% of synthetic facial images respectively.

Figure 5.15 illustrates the comparison of actual and estimated (median) value of θ_{yaw} of synthetic face moving from 0° to $+60^\circ$, $+60^\circ$ to -60° and back to 0° of θ_{yaw}. It

(a) HMOAAM (b) MOAAM (c) SOAAM

Figure 5.10: Localization of facial features of webcam facial images by HMOAAM, MOAAM and SOAAM

(a) HMOAAM (b) MOAAM (c) SOAAM

Figure 5.11: Localization of facial features of synthetic facial images by HMOAAM, MOAAM and SOAAM

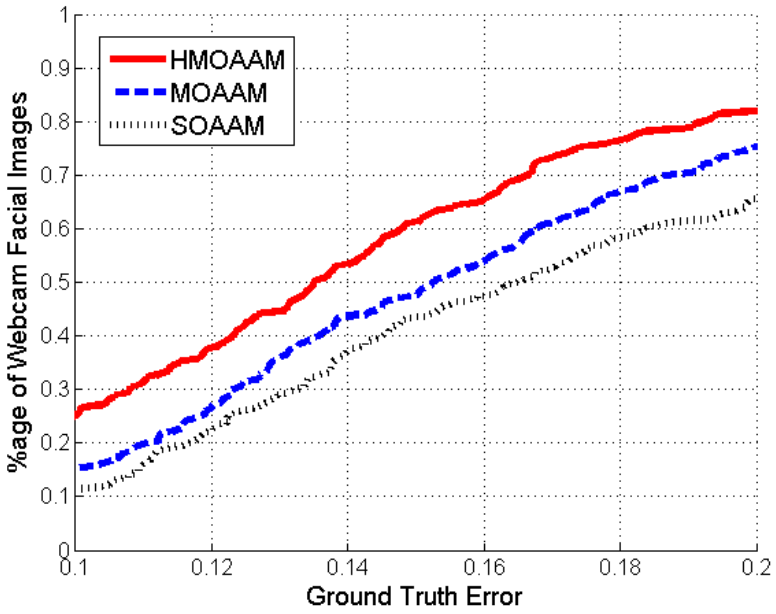


Figure 5.12: Ground truth error comparison of HMOAAM, MOAAM and SOAAM for webcam facial images

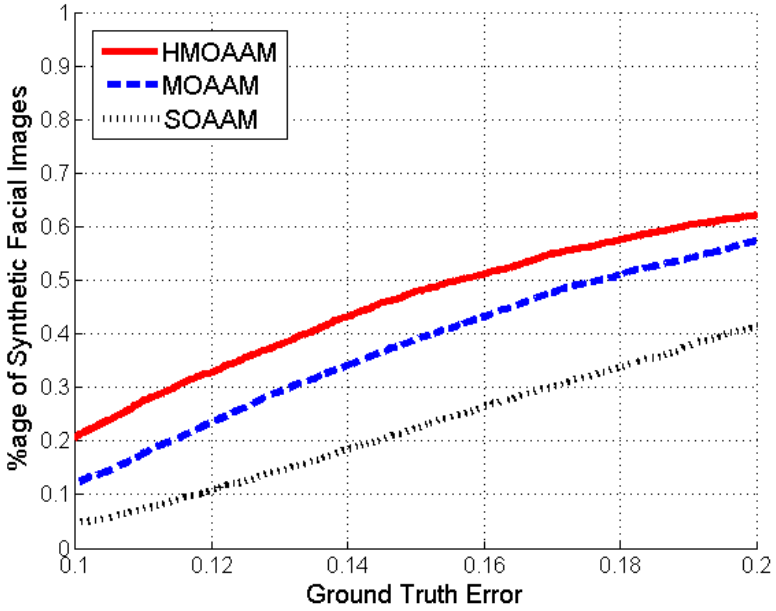


Figure 5.13: Ground truth error comparison of HMOAAM, MOAAM and SOAAM for synthetic facial images

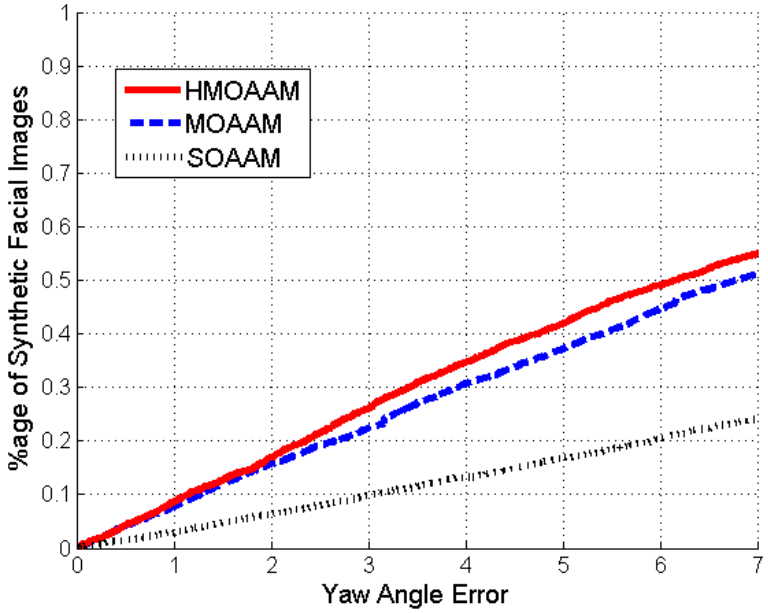


Figure 5.14: Comparison of error in θ_{yaw} by HMOAAM, MOAAM and SOAAM for synthetic facial images

can be analysed from the figure that the noise in the curve of the estimated yaw angle value around $\pm 60^{\circ}$ is due to the occlusion caused by the out-of-plane rotation of the face.

Table 5.3 illustrates the comparison of the all the methods for both the facial image databases. As far as time consumption is concerned, it is shown in the table 5.3 that HMOAAM requires 3000 warps to extract features of an oriented face. Single warp in an iteration is equal to 90% of the time consumed by an iteration i.e. 0.03 msec in Pentium-IV 3.2GHz. Thus for a complete facial analysis of a face HMOAAM requires 100 msec, which means it can successfully analyze 10 frames in one second. Other methods of MOAAM and SOAAM are also restricted to complete their facial search with in this time period in order to compare their robustness and efficiency with HMOAAM. SOAAM evolves for twice the number of generations than MOAAM and HMOAAM, since it analyzes single camera image whereas MOAAM and HMOAAM analyze double camera images. HMOAAM evolves for 14 generations compared to 15 in the case of MOAAM, to compensate time consumption of second phase of optimization by Gradient Descent.

It is important to point out that the above mentioned computational time is required to align the face without any prior knowledge of the facial pose, however in a face tracking mode the required time is reduced enormously by employing pose parameters of the previous frames, thus making it a true real time application.

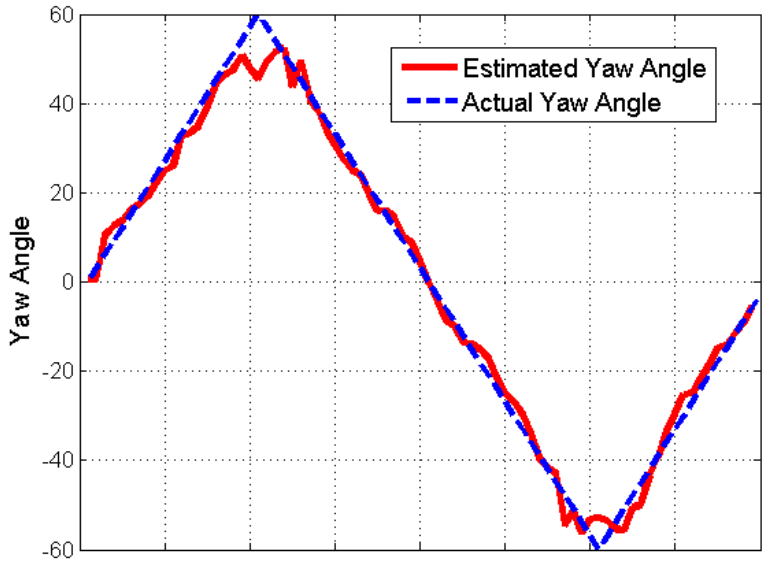


Figure 5.15: Comparison of actual and estimated θ_{yaw} of synthetic face moving laterally from 0° to +60° to −60° to 0°

Table 5.3: Analysis of the Results

Method	Database (Images)	Images Aligned ($GTE \leq 15\%$)	Pose Estimated ($E_{\theta_{yaw}} \leq 5°$)	pop x gen	No. of warps	Time (msec)
HMOAAM	Synthetic (4160)	**48%**	**42%**	100 x 14	3000	100
MOAAM	Synthetic (4160)	39%	37%	100 x 15	3000	100
SOAAM	Synthetic (4160)	22%	17%	100 x 30	3000	100
HMOAAM	Webcam (246)	**61%**	—	100 x 14	3000	100
MOAAM	Webcam (246)	48%	—	100 x 15	3000	100
SOAAM	Webcam (246)	43%	—	100 x 30	3000	100

5.5 Conclusions

This chapter presented a method to overcome the problem of pose estimation and facial features extraction by multiple cameras. In the algorithm of HMOAAM, we have presented a novel methodology of hybridization of NSGA-II with gradient descent for the real world optimization problem of facial analysis of multiple camera images by 2.5D Active Appearance Model (AAM). This hybridization provides a solution to efficiently tackle the problem of non-convexity of the multidimensional search space formed by face search in AAM.

For this hybridization we have proposed a new gradient operator in NSGA-II, which computes gradients of the error function in conjunction with the existing operator of mutation. Thus it does not increase the computational cost of the system. We have proposed a unique method of calculating the relevant facial information of each camera in multi-objective optimization which makes facial search optimization procedure efficient and robust. Our proposed algorithm HMOAAM is applied on number of real webcam images and synthetic facial images obtained from two cameras, and its results are compared with single view system (SOAAM) and a non-hybrid double camera system (MOAAM). Results illustrates that HMOAAM outperforms MOAAM and SOAAM in terms of efficiency, robustness and accuracy.

Chapter 6

Conclusions And Perspective

Contents

6.1 Conclusions

In this thesis work we have presented a real time solution for the face analysis in a cognitive radio equipment. A Cognitive Radio (CR) equipment is a radio device that, apart from the usual radio signal processing elements, also integrates a set of sensing capabilities for the CR support. According to the sensor's information, the CR will define the optimal configuration to give the best service. The constraints of the video sensor of a CR equipment are to extract facial identity, face orientation and facial features in order to guide the choice of the most adapted video and/or audio codec which must be used in order to optimize the quality of the communication taking into account the global context of the radio transmission. We have presented solutions to these problems faced by the video sensor of the CR equipment i.e. *"Pose estimation and facial features localization of unknown and oriented faces captured by the video sensor"*.

We have presented various model-based approaches and to tackle the face analysis problem, we chose to focus our work on the model-based method called Active Appearance Models (AAM). Therefore, our work has been made to make the active appearance model (AAM) more robust.

We first introduced the active appearance models and the number of its variations proposed by different researchers in the state of the art. We then prepared a map of the approaches proposed by different research teams to make the AAM more robust to variations of poses, expressions and identity. We have discussed three main directions of the research work to solve the problem stated in this thesis. It includes the approaches by extending the number of models, the approaches by fitting AAM on multi-view images and the approaches by improving the optimization methods for AAM.

We have seen that the previous methods did not presented the solutions to our problem since no approach considers the overall analysis of the poses and features of the unknown faces. Moreover, these methods generally lead to increased processing time.

In the first step, we have presented our contribution of 2.5D AAM, which is used throughout the experiments. Secondly we used this model to analyse the facial images of single camera system. Pose variability in an unknown face makes the feature extraction difficult for a classical AAM technique, which uses only deterministic method for the face search optimization. Therefore we proposed hybridization of direct search and deterministic methods for the optimization of face search as our second contribution. In single camera system we hybridized genetic algorithm (GA) and gradient descent (GD) method in a unique way to make the face analysis system more robust and efficient i.e. we have embedded GD inside GA to gain time efficiency. We compared this technique with GD working alone and another hybridization of GA with Simplex. Results validated our proposition.

Finally we used 2.5D AAM in multi-view system to overcome the problem of large lateral movements of a face. Our third contribution is the introduction of multi-objective optimization in AAM. For the images from multi-view system, a simultaneous face search optimization method of NSGA-II (a multi-objective version of GA) is introduced for 2.5D AAM. We hybridized it with GD to make the system robust and efficient to be implemented in real time. We tested the proposed algorithm in two databases and logged the results. By comparing the results of the single-objective AAM, simple multi-objective AAM and hybrid multi-objective AAM, we deduced that it is the best system for the pose estimation and feature localization of an unknown and oriented face. As it can be seen from the figure 6.1, which compares the results of the yaw angle absolute error of single and double camera system optimized by hybrid genetic algorithm, that the double cameras configuration is more robust to the large lateral movements of a face. Similarly for the GTE, figures 6.2 and 6.3 show the comparison between HMOAAM (chapter 5) and HGOAAM (chapter 4) algorithms with respect to synthetic and real webcam face databases respectively. Table 6.1 illustrates the values with which multiple camera's results are better than single especially in estimating the yaw angle of the test images.

It is important to point out that the above mentioned thesis work was done to

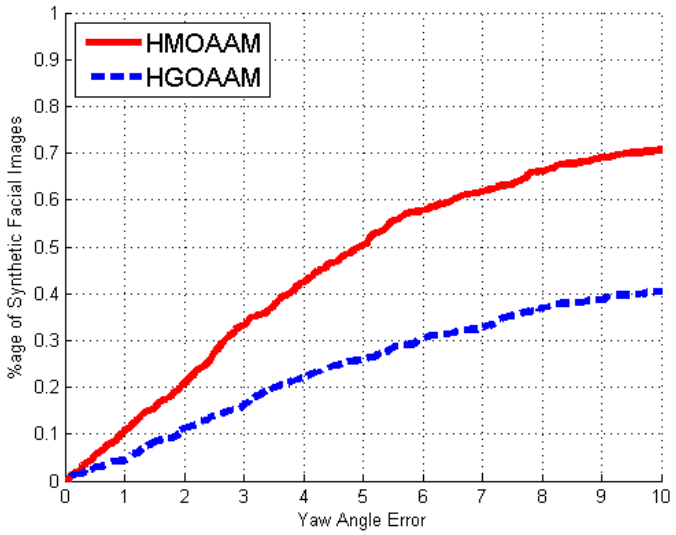


Figure 6.1: Robustness of yaw angle estimation with single and double camera

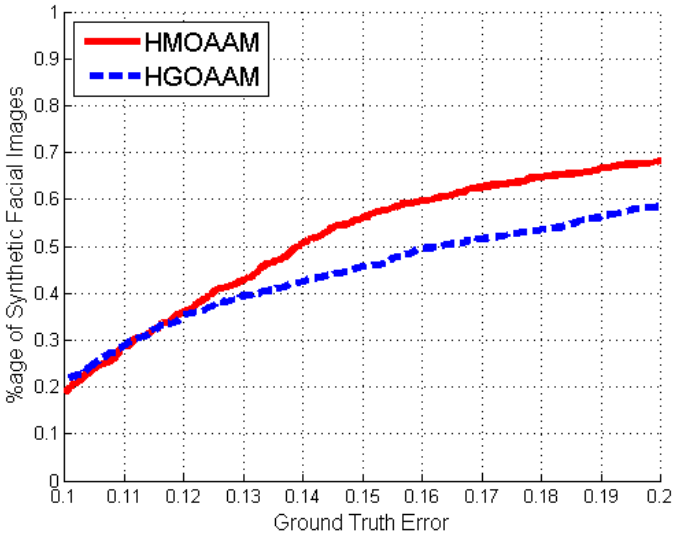


Figure 6.2: Ground truth error comparison of HMOAAM and HGOAAM for 600 Synthetic facial images.

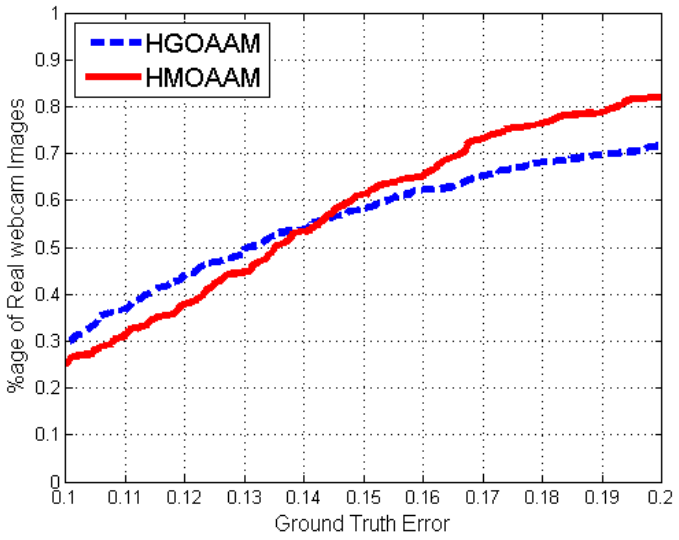


Figure 6.3: Ground truth error comparison of HMOAAM and HGOAAM for Real Webcam Database.

Table 6.1: Analysis of the Single and Multiple Camera Results

Method	Database (Images)	Images Aligned ($GTE \leq 15\%$)	Pose Estimated ($E_{\theta_{yaw}} \leq 5°$)	pop x gen	No. of warps	Time (msec)
HMOAAM	Synthetic (600)	**56%**	**50%**	100 x 14	3000	100
HGOAAM	Synthetic (600)	46%	26%	140 x 25	2500	83
HMOAAM	Webcam (246)	**61%**	—	100 x 14	3000	100
HGOAAM	Webcam (246)	58%	—	140 x 25	2500	83

align the face without any prior knowledge of the facial pose, however in face tracking mode the required time is reduced enormously by employing pose parameters of the previous frames, thus making it a true real time application. We have proposed a system different from the face trackers proposed by [27] by EBGM, [30] by simple GD and [53, 62, 68, 75, 76, 77, 84] by IC-LK algorithms. Our system can be applied not only as a face trackers but also other facial analysis tools like biometrics, face recognition related applications, acquiring and verifying passport databases and other various security related tools, which are concerned with highly accurate estimation of the facial pose and features without any of its prior knowledge in contrast to face tracking.

While performing the research work for this thesis we have published various research articles. Next section presents the details of these articles.

6.2 Publications

6.2.1 Journals

1. A. Sattar, R. Séguier, *HMOAM: Hybrid Multi-Objective Genetic Optimization for Facial Analysis by Appearance Model*, Journal of Memetic Computing (Springer Berlin / Heidelberg), Volume 2, Issue 1, pp. 25-46, march 2010, Springer Editions, DOI 10.1007/s12293-010-0038-3 ISSN 1865-9284 (Print) 1865-9292 (Online)

2. A. Sattar, N. Stoiber, R. Séguier, G. Breton, *Gamer's Facial Cloning for Online Interactive Games*, International Journal of Computer Games Technology (Hindawi), 2009

3. A. Sattar, R. Séguier, *HGOAAM: Facial Analysis by Active Appearance Model Optimized by Hybrid Genetic Algorithm*, Journal of Digital Information Management (JDIM), 2009

6.2.2 Conferences

1. A. Sattar, R. Séguier, *Hybrid Multi-Objective Active Appearance Model for Gamer's Facial Features Detection*, 22ème Colloque GRETSI , Dijon (France), September 8-10, 2009

2. A. Sattar, Y. Aïdarous, R. Séguier, *GAGM-AAM: A Genetic Optimization with Gaussian Mixtures for Active Appearance Models*, 15th International Conference on Image Processing (ICIP 2008), San Diego, California, USA, 12-15 October 2008

3. A. Sattar, R. Séguier, *MVAAM (Multi-View Active Appearance Model) Optimized by Multi-Objective Genetic Algorithm*, 8th International Conference on Automatic Face and Gesture Recognition (FG 2008), Amsterdam, The Netherlands, 17-19 September 2008

4. A. Sattar, R. Séguier, *GGA-AAM: Novel Heuristic Method of Gradient Driven Genetic Algorithm for Active Appearance Models*, Third International Conference on Digital Information Management (ICDIM 2008), London, United Kingdom, 13-16 November 2008

5. A. Sattar, Y. Aïdarous, S. Le Gallou, R. Séguier, *Face Alignment by 2.5D Active Appearance Model Optimized by Simplex*, 5th International Conference on Computer Vision Systems (ICVS2007), Bielefeld (Germany), March 21-24, 2007

6. Y. Aïdarous, S. Le Gallou, A. Sattar, R. Séguier, *Face Alignment using active appearance model optimized by simplex*, International Conference on Computer Vision Theory and Applications (VISAPP) Barcelona, Spain, March 2007

6.3 Applications

We have applied our proposition on two applications. First application is the part of the system in which the facial features are analysed by our proposition followed by its synthesis as an avatar on the display screen. A great effort was made by another PhD student of our team for the completion of this application. Second application uses 2.5D AAM for the orientation of a face in a computer game. Detailed description of these applications are given in the next sections.

6.3.1 Gamer's Facial Cloning

We have implemented MOAAM for a robust and efficient gamer's online cloning interactive system in [121] and shown in figure 6.4. Our system is composed of two cameras installed on the extreme edges of the screen to acquire real time images of the gamer. Gamer's face is analyzed and his pose and expressions are synthesized by the system to clone or retarget his features in the form of an avatar so that the gamers can interact with each other virtually. In the following sections we briefly explain our proposed interactive system stepwise.

6.3.1.1 Avatar's Face Modeling

In our proposed system, the visual aspect of the synthetic character is chosen by the user. Different classes of synthetic faces are available representing different ages, races, gender, physique and features etc. Once the class of the avatar is chosen, the required facial expressions, previously created in the system, are generated for this face. Note that the system's user has the possibility to edit the suggested facial expression to personalize the look of its avatar by manually clicking and moving the vertices. Ultimately the A-database (Avatar database) is created which contains the expressions, on the user-chosen character, which are necessary to form the A-Database.

We can build its appearance model according to the method presented in section 3.1 of chapter 3. This procedure delivers a reduced set of parameters which represent the principal variation patterns observed on the synthetic face (C_A). Manual marking of the

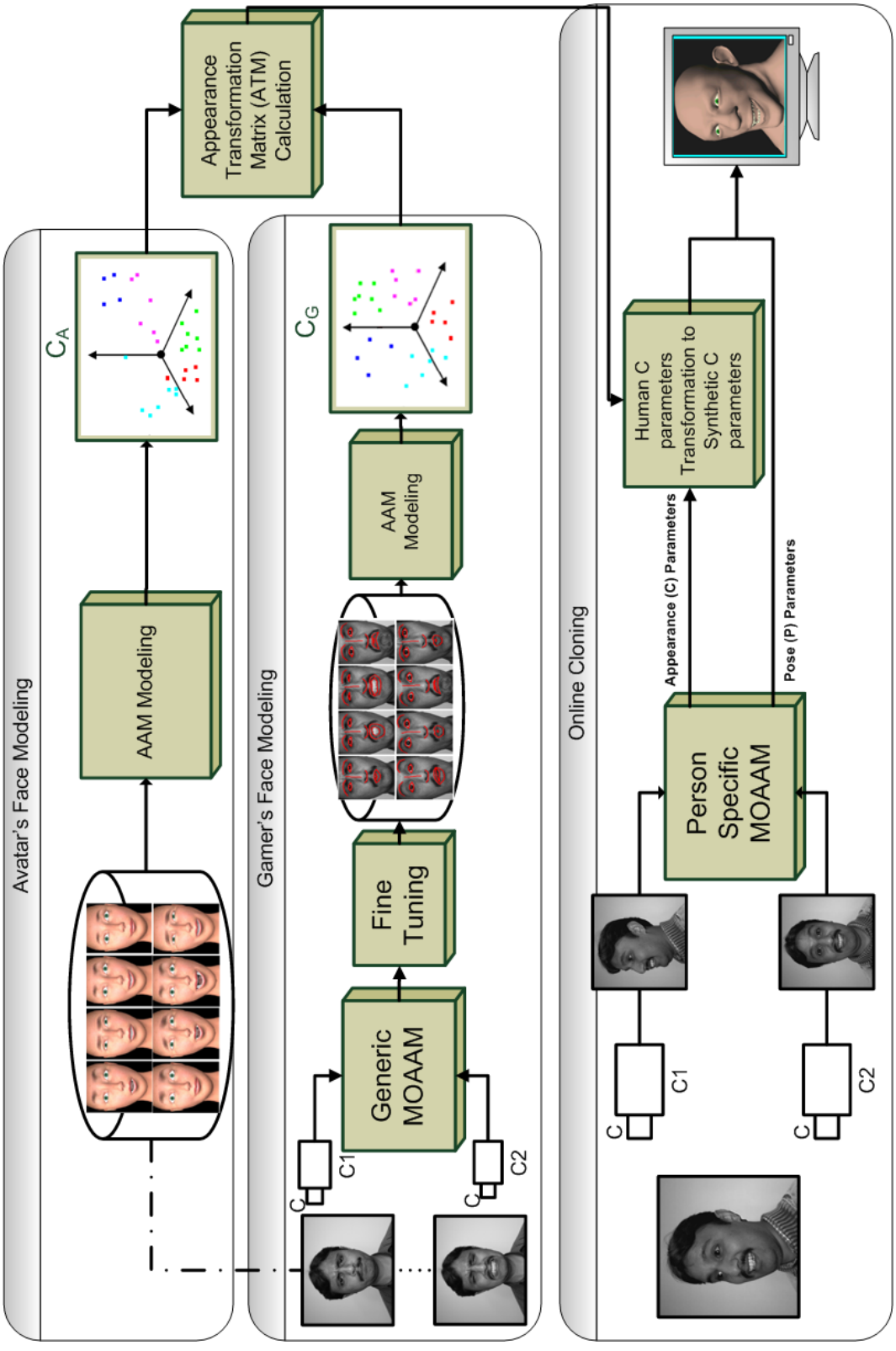


Figure 6.4: Block diagram of the gamer's facial cloning interactive system.

landmark on the synthetic face is not needed as the synthetic face is already generated by the system and it contains the location of each vertex.

6.3.1.2 Gamer's Face Modeling

Next step of our system is the creation of the gamer's face database similar to the one created for avatar. This is dome through a training procedure which is very simple and unproblematic. The essence of this phase is to make the system learn the facial deformations of the gamer's face so it can replicate the localization of features, emotions and gestures on the synthetic face. The gamer has to mimic the expressions that have an important impact on the formation of the appearance space.

In practice, the required facial expressions from previous section are displayed serially for the user to imitate. Facial images are captured by generic MOAAM, to automatically localize the facial features. Since user is unknown to the system therefore generic MOAAM containing an AAM model based on M2VTS facial images database is used. Feature localized by MOAAM is displayed on the screen for the user to fine tune the location of each feature. Finally all the facial images of the gamer are generated, each corresponding to synthetic facial expression of the A-Database. By reproducing these selected facial expressions of the gamer, we can build its very own appearance model along with its reduced appearance parameters C_G according to the method presented in section 3.1 of chapter 3. With C_G and C_A (obtained in previous section) we can calculate A_0 ATM (Appearance Transformation Matrix) mathematically. This ATM is capable of transforming the appearance parameters from the gamer's appearance space to the avatar's appearance space. It is gamer dependent and can be used for cloning only for particular gamer who was involved in generating it in the first place.

6.3.1.3 Online Cloning

From the previous two sections we obtained an ATM capable of transforming the appearance parameters from the gamer's appearance space to the avatar's appearance space. In online cloning, this transformation involves only a matrix multiplication of real time gamer's appearance parameters C_G with A_0 to obtain avatar's appearance parameters C_A. This analytically simple framework enables real-time performances. The virtual illustration of a gamer is cloned in the form of an avatar synthesized by C_A and ultimately display on the screen as shown on figure 6.4.

The appearance parameters of a gamer are acquired in real time by our facial analysis system of multiple cameras. Tactical moves of the game causes the gamer to move a lot in different direction. Employing multiple cameras resolved this problem. Two cameras placed at the extreme edges of the screen acquire real time image of the gamer and at the same time his facial features and pose are analyzed by person specific MOAAM. Person specific MOAAM model which is generated from the gamer database of the previous section, contains all the *pose-free* facial variations of the gamer. User's oriented face is analysed by this MOAAM, to give its appearance and pose parameters. Appearance parameters are transformed by ATM in the synthetic face's parameter space and

synthetic face is synthesized by them. After that pose parameters are used to adjust the orientation of the avatar being displayed on the screen. As shown in the cloning section of the figure 6.4, appearance parameters undergoes transformation while pose parameter are directly reproduced on the avatar face to clone both the gamer's expressions and gestures. The linearity of the AAM scheme allows the reproduction of both extreme and intermediate facial expressions and movements, with low computing requirements.

A reduced version of this application was presented in a demo held at SUPELEC in may 2009. The demo was named as "Journée des Images et Système Embarquée" (JISE). Figure 6.5 shows the snapshot of that system.

Figure 6.5: Facial Cloning system (Courtesy TF3)

6.3.2 Demo (Rennes Atlante)

Second application of our proposition was made for the demonstration shown to the public in the exhibition held in "Rennes Atalante, Destination High Tech". This application uses 2.5D AAM in single-view system. A demonstration was prepared in the form of a game called "Mosquito Eater". In the beginning of the game, gamer is requested to stay in a frontal view for couple of seconds. In this first step appearance parameters of the unseen gamer is logged with the help of a generic 2.5D AAM. This generic AAM model analyses the gamers face and finds the closest match of the gamer's facial appearance. When the system is converged, appearance parameters are locked and are not varied until new gamer is introduced. In the gaming phase only pose parameters are varied, hence the application becomes person specific. By evaluating the pose parameters or the orientation of the gamer's face in real time, i.e. the face's rotational and translational parameters, a cursor is accordingly moved on the display screen in

order to shoot mosquitoes with a button. Each successful mosquito catch increases the score of the gamer. A snapshot of the system is shown in the figure 6.6.

Figure 6.6: Snapshot of Mosquito Eater computer game

6.4 Perspectives

Some research perspectives appear at the end of this thesis. First of all we discuss the problems faced in single view AAM. No matter how fast the direct search methods become they remain slower than any deterministic method used alone, but at the same time they can be more robust. Therefore we have combined both of them to reach a robust real time system. We have a trade off between efficiency and robustness while using combination of two methods, by increasing the deterministic approach of our algorithm system becomes more time efficient and less robust and vice versa.

Additionally we have not implemented various techniques to compensate the illumination variations in the face analysis. We have simply applied photometric normalization of the texture. Since 2.5D AAM was build from M2VTS database which does not include illumination variations, therefore we did not face huge problems in the testing phase. Illuminations problem can be solved globally or locally. Globally, we can apply different illumination correction methods of histogram equalization like CLAHE (Contrast Limited Adaptive Histogram Equalization) [32, 122]. Locally, as in 3DMM the model contains the 3D coordinates of each pixel, with which they can create the virtual lighting in the model. Similar virtual lighting can be created in our 2.5D AAM.

Number of points and triangles in the shape model can be increased to increase the robustness of the system. Region around cheeks and the chin do not deform a lot in the expression variations hence they can be neglected and the size of the AAM model can

be reduced. Moreover the background neighboring the cheeks and the chin makes the error function's curve more complex, making the facial search to fall in local minima. By considering the facial region only around eyebrows, eyes and mouth we can neglect the effects of the background. Additionally, this reduction in the size of the model would also increase the rapidity of the system. On the other hand information of the ear can be incorporated inside the texture to include additional facial information in the profile and semi profile views. This expansion of the model and the texture increases the robustness of the system while decreasing its speed.

On the deepening of our work on the AAM, experiments on the variability in expression are to be implemented. Indeed, we showed the relevance of our system to the variability in identity, pose and features localization. Furthermore, in order to monitor the faces expressions, work continues in the laboratory SCEE-SUPELEC/IETR on Active Appearance Models.

In multi camera system, calibration of the cameras is one of the main problem. A stereo rig can be installed on the display screen to minimize the calibration problem, otherwise user has to calibrate the cameras manually. The problem of multiple processing instead of only single image processing drastically reduces the time efficiency of the system. Although by calculating the CIRF one can control and switch off the undesirable camera but still it require twice the processing times. This cannot be seen as a drawback of the system, because by using multiple cameras robustness of the system against out-of-plane rotation of the face increases. Work is being done in the domain of GPGPU (General Purpose Graphical Processing Unit) to increase the speed of the multi camera system by increasing the processing power. It is due to the parallel nature of the direct search algorithm, used in this thesis, that makes them to be implemented in GPGPU.

For the moment, this multi camera approach is tested in a double camera system, but it would be interesting to extend it for larger events, like conferences and meetings, with multiple cameras installed on different corners of the room and displayed on video projectors.

Appendix A

NSGA-II

This section presents the detailed description of the algorithm of NSGA-II by [6].

A.1 A fast non-dominated sorting approach

In order to sort a population according to the level of non-domination, each solution must be compared with every other solution in the population to find if it is dominated. This requires comparisons for each solution with respect to the number of objectives. This process is continued to find the members of the first nondominated class for all population members. In order to find the individuals in the next front, the solutions of the first front are temporarily discounted and the above procedure is repeated.

In the following we describe a fast non-dominated sorting approach proposed by [6] which will require less computations. First, for each solution we calculate three entities: (1) $S[ii]$, the indexes of the solutions which dominate the solution ii, (2) $Q[ii]$, a count of solutions dominated by ii and (3) $Z[ii]$ a count of the solutions which have dominated ii. The ranking algorithm is presented in the following pseudo code.

Algorithm A.1.1: NON DOMINATING SORT$(PopSize, Error)$

$$
\begin{array}{l}
\textbf{for } ii \leftarrow 1 \textbf{ to } PopSize \\
\quad \textbf{do} \begin{cases} \textbf{for } jj \leftarrow ii+1 \textbf{ to } PopSize \\ \quad \textbf{do} \begin{cases} \textbf{if } Error[ii] < Error[jj] \\ \quad \textbf{then} \begin{cases} S[ii, Q[ii]] \leftarrow jj \\ Q[ii] \leftarrow Q[ii]+1 \\ Z[jj] \leftarrow Z[jj]+1 \end{cases} \\ \textbf{else if } Error[ii] > Error[jj] \\ \quad \textbf{then} \begin{cases} S[jj, Q[jj]] \leftarrow ii \\ Q[jj] \leftarrow Q[jj]+1 \\ Z[ii] \leftarrow Z[ii]+1 \end{cases} \end{cases} \end{cases} \\
\textbf{return } (S, Q, Z)
\end{array}
$$

$S[ii]$ contains the indexes of all the solutions which are dominated by solution ii. For example if a solution is to be ranked as the first rank it would contain indexes of all the solutions. But since it cannot contain other solutions of the first rank therefore we cannot verify the rank considering only this parameter. $Q[ii]$ show the count of these solutions that ii have dominated. $Z[ii]$ show that how many solutions have dominated ii. For example if it contains zero it means that it is of first rank. Therefore solution of the first rank are extracted by this variable according to

Algorithm A.1.2: EXTRACTING FIRST RANK$(PopSize, Z)$

$$
\begin{array}{l}
nif \leftarrow 1 \\
\textbf{for } kk \leftarrow 1 \textbf{ to } PopSize \\
\quad \textbf{do} \begin{cases} \textbf{if } Z[kk] = 0 \\ \quad \textbf{then} \begin{cases} ParetoFront[1, nif] \leftarrow kk \\ nif \leftarrow nif + 1 \end{cases} \end{cases} \\
NumIndsFront[1] \leftarrow nif - 1; \\
\textbf{return } (ParetoFront[1], NumIndsFront[1])
\end{array}
$$

where $ParetoFront_{(1,nif)}$ contains solutions of the first rank and $NumIndsFront_{(1)}$ represent the count of these solutions. Until now the first rank has been assigned to the solutions. Starting from the first rank we will take each solution of the first rank and find out the solutions from which it was dominated. Afterwards we decrease the $Z[ii]$ of each solution by one as the solutions of the first rank are now removed from the population. And also check the $Z[ii]$ of these solution whether it is reached zero or not, if yes then we have the solutions of the second rank. This procedure is repeated until there are no further solutions to be ranked. The procedure is presented below.

Algorithm A.1.3: RANKING$(S, Q, Z, ParetoFront[1], NumIndsFront[1])$

$$
\begin{array}{l}
fid \leftarrow 1 \\
\textbf{while } NumIndsFront[fid] \neq 0 \\
\quad \textbf{do} \begin{cases} nif \leftarrow 1 \\ \textbf{for } ii \leftarrow ParetoFront[fid, 1] \textbf{ to } ParetoFront[fid, end] \\ \quad \textbf{do} \begin{cases} \textbf{for } jj \leftarrow 1 \textbf{ to } Q[ii] \\ \quad \textbf{do} \begin{cases} VerifInd \leftarrow S[ii, jj] \\ Z[VerifInd] \leftarrow Z[VerifInd] - 1 \\ \textbf{if } Z[VerifInd] = 0 \\ \quad \textbf{then} \begin{cases} ParetoFront[fid + 1, nif] \leftarrow VerifInd \\ nif \leftarrow nif + 1 \end{cases} \end{cases} \end{cases} \\ fid \leftarrow fid + 1 \\ NumIndsFront[fid] \leftarrow nif - 1 \end{cases} \\
\textbf{return } (ParetoFront, NumIndsFront)
\end{array}
$$

A.2 Crowding Distance

To get an estimate of the density of solutions surrounding a particular point in the population we take the average distance of the two points on either side of this point along each of the objectives. This quantity $i_{distance}$ serves as an estimate of the size of the largest cuboid enclosing the point i without including any other point in the population (we call this the crowding distance). In figure A.1, the crowding distance of the *ith* solution in its front (marked with solid circles) is the average side-length of the cuboid (shown with a dashed box).

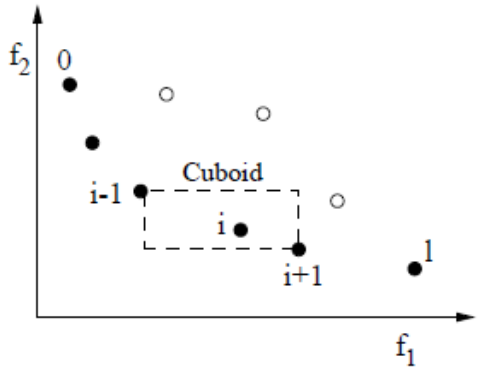


Figure A.1: Crowding distance calculation

Let us assume that every individual i in the population has now two attributes (i) Non-domination rank (ii_{rank}) and (ii) Local crowding distance ($ii_{distance}$). We can define a sort by

$$f_i(ii) \geq f_i(jj),\ if(ii_{rank} < jj_{rank}) or((ii_{rank} = jj_{rank}) and(ii_{distance} > jj_{distance})) \tag{A.1}$$

That is, between two solutions with differing non-domination ranks we prefer the point with the lower rank. Otherwise, if both the points belong to the same front then we prefer the point which is located in a region with lesser number of points (the size of the cuboid enclosing it is larger).

A.3 The Main Loop

Initially, a random parent population P[t] is created at t=0. Fitness of each solution is calculated and they are sorted based on the non-domination sort (algorithms A.1.1, A.1.2 and A.1.3). Each solution is assigned a rank equal to its non-domination level (1 is the best level). Tournament selection, crossover, and mutation operators are used to create a child population Q[t] of size N.

Secondly, a combined population $R[t] = P[t] \cup Q[t]$ is formed. The population R[t] will be of size 2N. Then, the population R[t] is sorted according to non-domination. The new parent population P[t+1] is formed by adding solutions from the first front till the size exceeds N. Thereafter, the solutions of the last accepted front are sorted according to the crowding distances (statement A.1) and the first N points are picked. This population of size N is now used for selection, crossover and mutation to create a new population Q[t+1] of size N. This process is repeated for particular number of generation until the stopping criteria of the algorithm is met. Pseudo code of the algorithm is present below.

Algorithm A.3.1: MAIN LOOP()

$P[t]$ is created randomly
$Nondominatingsort(P[t])$
while stopping criteria is not met
do $\begin{cases} Q[t] \leftarrow Reproduction(P[t]) \\ R[t] \leftarrow P[t] \cup Q[t] \\ NondominatingSort(R[t]) \\ CrowdingDistance(R[t]) \\ P[t+1] \leftarrow R[t, 0:N] \\ t \leftarrow t+1 \end{cases}$

Appendix B

List of Acronyms and Abbreviations

2.5D AAM	**2.5 D**imension **A**ctive **A**pperance **M**odel
3D AMB AAM	**3 D**imension **A**nthropometry **M**uscle **B**ased **A**ctive **A**pperance **M**odel
3DMM	**3 D**imension **M**orphable **M**odel
AAM	**A**ctive **A**pperance **M**odel
ASM	**A**ctive **S**hape **M**odel
ATM	**A**ppearance **T**ransformation **M**atrix
AWN	**A**ctive **W**avelet **N**etwork
CCA	**C**anonical **C**orrelation **A**nalysis
CELP	**C**ode **E**xcited **L**inear **P**redictive
CIRF	**C**amera **I**nformation **R**elevance **F**actor
CLAHE	**C**ontrast **L**imited **A**daptive **H**istogram **E**qualization
CNN	**C**onvolution **N**eural **N**etworks
COG	**C**enter **O**f **G**ravity
CR	**C**ognitive **R**adio
CRM	**C**ognitive **R**adio **M**anagement
CT	**C**omputed **T**omography
CV-AAM	**C**oupled **V**iew **A**ctive **A**pperance **M**odel
DAM	**D**irect **A**pperance **M**odel
DCT	**D**iscrete **C**osine **T**ransform
DIRECT	**DI**viding **RECT**angles
DOF	**D**egree **O**f **F**reedom
EBGM	**E**lastic **B**unch **G**raph **M**atching
FAM	**F**lexible **A**pperance **M**odel
FBG	**F**ace **B**unch **G**raphs
FOV	**F**ield **O**f **V**iew
GA	**G**enetic **A**lgorithm
GA-Simplex	**G**enetic **A**lgorithm **Simplex**
GAGD	**G**enetic **A**lgorithm **G**radient **D**escent
GAGM-AAM	**G**enetic **A**lgorithm with **G**aussian **M**ixture for

	Active **A**pperance **M**odel
GGA-AAM	**G**radient driven **G**enetic **A**lgorithm for **A**ctive **A**pperance **M**odel
GHz	**G**iga **H**ertz
GPU	**G**raphical **P**rocessing **U**nit
GPGPU	**G**eneral **P**urpose **G**raphical **P**rocessing **U**nit
GRETSI	**G**roupe de **R**echerche et d'**E**tudes du **T**raitement de **S**ignal et des **I**mages
GTE	**G**round **T**ruth **E**rror
GWT	**G**abor **W**avelet **T**ransform
HGA-Simplex	**H**ybrid **G**enetic **A**lgorithm **Simplex**
HGOAAM	**H**ybrid **G**enetic **O**ptimization for **A**ctive **A**pperance **M**odel
HMOAAM	**H**ybrid **M**ultiobjective **O**ptimization for **A**ctive **A**pperance **M**odel
HMOO	**H**ybrid **M**ultiobjective **O**ptimization
IDEA	**I**terated **D**ensity **E**stimation **E**volutionary **A**lgorithms
IC-LK	**I**nverse **C**ompositional **L**ucas *K*anade
JISE	**J**ournée **I**mage et **S**ystèmes **E**mbarqués
KPCA	**K**ernel **P**rincipal **C**omponent **A**nalysis
LSM	**L**ocal **S**earch **M**ethod
M2VTS	**M**ulti **M**odal **V**erification for **T**eleservices and **S**ecurity applications
MAYA	is a high-end 3D computer graphics and 3D modeling software package
MC	**M**onte **C**arlo
MOAAM	**M**ultiobjective **O**ptimization for **A**ctive **A**pperance **M**odel
MOEA	**M**ulti **O**bjectvie **E**volutionary **A**lgorithm
MOGA	**M**ulti **O**bjectvie **G**enetic **A**lgorithm
MRI	**M**agnetic **R**esonance **I**maging
MVAAM	**M**ulti **V**iew **A**ctive **A**pperance **M**odel
NPGA	**N**iched **P**areto **G**enetic **A**lgorithm
NSGA	**N**on-dominated **S**orting **G**enetic **A**lgorithm
NSGA-II	**N**on-dominated **S**orting **G**enetic **A**lgorithm **II**
OSI	**O**pen **S**ystem **I**nterconnection
PCA	**P**rincipal **C**omponent **A**nalysis
PDM	**P**oint *D*istribution *M*odel
POSIT	**P**ose **O**rthography and **S**caling with **IT**eration
PPBTF	**P**ixel **P**attern **B**ased **T**exture **F**eature
RANSAC	**RAN**dom **SA**mple **C**onsensus
RBF	**R**adial **B**asis **F**unction
RS	**R**andom **S**earch
RW	**R**andom **W**alk
SA	**S**imulated **A**nnealing

SCEE	**S**ignal, **C**ommunication et **E**lectronique **E**mbarquée
SD	**S**teepest **D**escent
SDR	**S**oftware **D**efined **R**adio
SOAAM	**S**ingleobjective **O**ptimization for **A**ctive **A**pperance **M**odel
SQP	**S**equential **Q**uadratic *P*rogramming
STAAM	**ST**ereo **A**ctive **A**ppearance **M**odel
SUPELEC	L'Ecole **SUP**érieure d'**ELEC**tricité
SVM	**S**upport **V**ector **M**achine
TB-AAM	**T**ensor **B**ased **A**ctive **A**pperance **M**odel
TC-ASM	**T**exture **C**onstrained **A**ctive **S**hape **M**odel
TS	**T**abu **S**earch
TST	**T**emplate **S**election **T**racker
TSWOR	**T**ournament **S**election **W**ith**O**ut **R**eplacement
Twin-VQ	**T**ransform-domain **W**eighted **IN**terleaved **V**ector **Q**uantization
VOD	**V**ideo **O**n **D**emand

Bibliography

[1] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision*, volume 2, pages 484–498, 1998.

[2] N. Cladel and R. Séguier. Active contours multiobjective optimization by hybrid algorithm. In *Irish Machine Vision and Image Processing Conference 2004*, 2004.

[3] M. B. Stegmann. Active appearance models: Theory, extensions and cases. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, aug 2000. URL `http://www.imm.dtu.dk/~aam/main/`.

[4] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal Royal Statistical Society*, pages 53:285–339, 1991.

[5] N. Durand and J. Alliot. A combined nelder-mead simplex and genetic algorithm. *The Genetic and Evolutionary Computation Conference*, pages 1–7, 1999.

[6] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Parallel Problem Solving from Nature VI Conference*, pages 849–858, 2000.

[7] J. Mitola. *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Royal Inst. of Tech., Sweden, 2000.

[8] V. Vezhnevets, S. Soldatov, and A. Degtiareva. Automatic extraction of frontal facial features. In *ACCV'04, Asian Conference on Computer Vision*, volume 2, pages 1020–1025, 2004.

[9] T. Kawaguchi and M. Rizon. Iris detection using intensity and edge information. *Pattern Recognition*, 36(2):549–562, 2003.

[10] W. Lu, H.and Zhang and D. Yang. Eye detection based on rectangle features and pixel-pattern-based texture features. In *Proc. International Symposium on Intelligent Signal Processing and Communication Systems ISPACS 2007*, pages 746–749, 2007. doi: 10.1109/ISPACS.2007.4445995.

[11] T. Lee, S. K. Park, and M. Park. A new facial features and face detection method for human-robot interaction. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2005*, pages 2063–2068, 2005.

[12] T. Lee, S.K. Park, and M. Park. Novel pose-variant face detection method for human-robot interaction application. In *IAPR Conference on Machine Vision Applications*, pages 281–284, 2005.

[13] J. Wu and Z.H. Zhou. Efficient face candidates selector for face detection. *Pattern Recognition*, 36(5):1175–1186, 2003.

[14] V. Kruger, A. Happe, and G. Sommer. Affine real-time face tracking using gabor wavelet networks. In *Proc. 15th International Conference on Pattern Recognition*, volume 1, pages 127–130 vol.1, 2000. doi: 10.1109/ICPR.2000.905289.

[15] T. E. Campos R. S. Feris and R. M. Cesar Jr. A project for face recognition from video sequences using GWN and eigenfeature selection. In *In Proceedings of WAICV'2000 Workshop on Artificial Intelligence and Computer Vision*, pages 141–145, 2000.

[16] L. V. Praseeda and Dr. M. Sasikumar. A neural network based facial expression analysis using gabor wavelets. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 32, pages 569–573, 2008.

[17] S. Duffner and C. Garcia. Robust face alignment using convolutional neural networks. In *Proceedings of the Internation Conference on Computer Vision Theory and Applications (VISAPP)*, Funchal, Portugal, January 2008.

[18] N. Gourier, D. Hall, and J. L. Crowley. Facial features detection robust to pose, illumination and identity. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 617–622 vol.1, 2004. doi: 10.1109/ICSMC.2004.1398368.

[19] I. O. Stathopoulou and G. A. Tsihrintzis. An improved neural-network-based face detection and facial expression classification system. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 666–671 vol.1, 2004. doi: 10.1109/ICSMC.2004.1398376.

[20] P.D. Sozou, T.F. Cootes, and C.J. Taylor. A non-linear generalisation of point distribution models using polynomial regression. In *British Machine Vision Conference*, pages 397–406, 1994.

[21] T.F. Cootes and C.J. Taylor. Active shape model search using local grey-level models: A quantitative evaluation. In *British Machine Vision Conference*, pages 639–648, 1993.

[22] L. Wiskott, J. M. Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, 19(7):775–779, 1997. ISSN 0162-8828. doi: 10.1109/34.598235.

[23] A. Lanitis, C. J. Taylor, and T. F. Cootes. An automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13:393–401, 1995.

[24] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH'99*, pages 187–194, 1999.

[25] J. Ahlberg. Candide-3 - an updated parameterised face. Technical Report LiTH-ISY-R-2326, Image Coding Group, Dept. of Electrical Engineering, Linkoping University, Sweden, 2001.

[26] M. Rydfalk. Candide, a parameterized face. Technical Report LiTH-ISY-I-866, Dept. of Electrical Engineering, Linköping University,, 1987.

[27] T. Maurer and C. von der Malsburg. Tracking and learning graphs and pose on image sequences of faces. In *Proc. Second International Conference on Automatic Face and Gesture Recognition*, pages 176–181, 1996. doi: 10.1109/AFGR.1996.557261.

[28] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *European Conference on Computer Vision*, 2008. `http://www.milbo.users.sonic.net/stasm`.

[29] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003. ISSN 0162-8828. doi: 10.1109/TPAMI.2003.1227983.

[30] F. Dornaika and J. Ahlberg. Fast and reliable active appearance model search for 3D face tracking. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34:1838–1853, 2004.

[31] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 300–305, 14–16 April 1998. doi: 10.1109/AFGR.1998.670965.

[32] S. Le Gallou, G. Breton, C. Garcia, and R. Seguier. Distance maps: A robust illumination preprocessing for active appearance models. In *VISAPP06*, 2006.

[33] T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision, 2004. Technical report from Imaging Science and Biomedical Engineering, University of Manchester.

[34] 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, June 2001. doi: 10.1109/34.927467.

[35] T. F. Cootes and C. J. Taylor. Constrained active appearance models. In *Proc. Eighth IEEE International Conference on Computer Vision ICCV 2001*, volume 1, pages 748–754 vol.1, 2001. doi: 10.1109/ICCV.2001.937601.

[36] M. B. Stegmann. *Generative Interpretation of Medical Images*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004. URL `http://www2.imm.dtu.dk/pubdb/p.php?3126`. Awarded the Nordic Award for the Best Ph.D. Thesis in Image Analysis and Pattern Recognition in the years 2003-2004 at SCIA'05.

[37] T.F. Cootes and C.J. Taylor. An algorithm for tuning an active appearance model to new data. In *British Machine Vision Conference*, volume 3, page III:919, 2006.

[38] T.F. Cootes, G.J. Edwards, and C.J. Taylor. A comparative evaluation of active appearance model algorithms. In *British Machine Vision Conference*, volume 2, pages 680–689, 1998.

[39] T.F. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. In *British Machine Vision Conference*, volume 2, pages 837–846, 2002.

[40] X. Hou, S. Z. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001*, volume 1, pages I–828–I–833 vol.1, 2001. doi: 10.1109/CVPR.2001.990568.

[41] S. Romdhani, A. Psarrou, and S. Gong. On utilising template and feature-based correspondence in multiview appearance models. In *Sixth European Conference on Computer Vision*, volume 1, pages 799–813, 2000.

[42] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[43] S. Yan, C. Liu, S.Z. Li, H. Zhang, H.Y. Shum, and Q. Cheng. Face alignment using texture-constrained active shape models. *Image and Vision Computing*, 21: 69–75, 2003.

[44] H.S. Lee and D. Kim. Illumination-robust face recognition using tensor-based active appearance model. In *8th IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[45] M. Alex, O. Vasilescu, and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *In Proceedings of the European Conference on Computer Vision*, pages 447–460, 2002.

[46] M. Alex, O. Vasilescu, and D. Terzopoulos. Multilinear image analysis for facial recognition. In *Proceedings of the International Conference on Pattern Recognition*, 2002.

[47] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60:135–164, 2004.

[48] H. Mercier, J. Peyras, and P. Dalle. Toward an efficient and accurate AAM fitting on appearance varying faces. In *Proc. 7th International Conference on Automatic Face and Gesture Recognition FGR 2006*, pages 363–368, 2006. doi: 10.1109/FGR.2006.104.

[49] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 3:221–225, 2004.

[50] C. Hu, R. Feris, and M. Turk. Active wavelet networks for face alignment. In *British Machine Vision Conference*, 2003.

[51] T.F. Cootes, K.N. Walker, and C.J. Taylor. View-based active appearance models. In *FGR'00, International Conference on Automatic Face and Gesture Recognition*, pages 227–232, 2000.

[52] T. Shan, B.C. Lovell, and S. Chen. Face recognition robust to head pose from one sample image. *18th International Conference on Pattern Recognition (ICPR'06)*, pages 515–518, 2006.

[53] J. Sung and D. Kim. Pose-robust facial expression recognition using view-based 2D+3D AAM. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 38(4):852–866, 2008. ISSN 1083-4427. doi: 10.1109/TSMCA.2008.923047.

[54] S. Z. Li, Y. Shuicheng, H. Zhang, and Q. Cheng. Multi-view face alignment using direct appearance models. In *Proc. Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 324–329, 2002. doi: 10.1109/AFGR.2002.1004174.

[55] S. Xin and H. Ai. Face alignment under various poses and expressions. *Affective Computing and Intelligent Interaction, First International Conference, ACII 2005, Beijing, China*, 3784, 2005.

[56] Y. Liu, Y. Li, L. Tao, and G. Xu. Multi-view face alignment guided by several facial feature points. *Proceedings of the Third International Conference on Image and Graphics, ICIG'04*, pages 238–241, 2004.

[57] X. Feng, B. Lv, and Z. Li. Automatic facial expression recognition using both local and global information. In *Chinese Control Conference*, pages 1878–1881, 2006.

[58] M. Romero and A.F. Bobick. Tracking head yaw by interpolation of template responses. *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, 5:83, 2004.

[59] J. Peyras, A.E. Bartoli, and S.K. Khoualed. Pools of AAMs: Towards automatically fitting any face image. In *British Machine Vision Conference*, pages xx–yy, 2008.

[60] J. Yang and H. Byun. Multi subspaces active appearance models. In *International Conference on Automatic Face and Gesture Recognition*, 2008.

[61] C. Hu, R. Feris, and M. Turk. Real-time view-based face alignment using active wavelet networks. In *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures AMFG 2003*, pages 215–221, 2003.

[62] J. Sung and D. Kim. Extension of AAM with 3D shape model for facial shape tracking. *International Conference on Image Processing, ICIP04*, 5:3363 – 3366, 2004.

[63] S. Von Duhn, L. Yin, M. J. Ko, and T. Hung. Multiple-view face tracking for modeling and analysis based on non-cooperative video imagery. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07*, pages 1–8, 2007. doi: 10.1109/CVPR.2007.383519.

[64] J. Paterson and A. Fitzgibbon. 3D head tracking using non-linear optimization. *British Machine Vision Conference*, 2003. URL `http://citeseer.ist.psu.edu/672545.html`.

[65] R. Ishiyama, M. Hamanaka, and S. Sakamoto. An appearance model constructed on 3D surface for robust face recognition against pose and illumination variations. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(3):326–334, 2005. ISSN 1094-6977. doi: 10.1109/TSMCC.2005.848193.

[66] S. Malassiotis and M. G. Strintzis. Real-time head tracking and 3D pose estimation from range data. In *Proc. International Conference on Image Processing ICIP 2003*, volume 2, pages II–859–62 vol.3, 2003. doi: 10.1109/ICIP.2003.1246816.

[67] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2D+3D active appearance models. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR04*, 2:II–535 – II–542, 2004.

[68] C. Hu, J. Xiao, I. Matthews, S. Baker, J.F. Cohn, and T. Kanade. Fitting a single active appearance model simultaneously to multiple images. *British Machine Vision Conference*, 2004.

[69] S. Koterba, S. Baker, I. Matthews, C. Hu, J. Xiao, J. Cohn, and T. Kanade. Multi-view AAM fitting and camera calibration. In *Proc. Tenth IEEE International Conference on Computer Vision ICCV 2005*, volume 1, pages 511–518 Vol. 1, 2005. doi: 10.1109/ICCV.2005.157.

[70] K. Ramnath, S. Koterba, J. Xiao, C. Hu, I. Matthews, S. Baker, J. Cohn, and T. Kanade. Multi-view AAM fitting and construction. *International Journal of Computer Vision*, 2007.

[71] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *In European Conference on Computer Vision*, pages 573–587, 2004.

[72] M. D. Cordea and E. M. Petriu. A 3D anthropometric-muscle-based active appearance model. *IEEE Transactions on Instrumentation and Measurement*, 55 (1):91–98, 2006. ISSN 0018-9456. doi: 10.1109/TIM.2005.860861.

[73] P. Martins and J. Batista. Single view head pose estimation. In *Proc. 15th IEEE International Conference on Image Processing ICIP 2008*, pages 1652–1655, 2008. doi: 10.1109/ICIP.2008.4712089.

[74] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylors. Coupled-view active appearance models. *Proceedings of British Machine Vision Conference*, 1: 52–61, 2000.

[75] D. Kim and J. Sung. A real-time face tracking using the stereo active appearance model. In *Proc. IEEE International Conference on Image Processing*, pages 2833–2836, 2006. doi: 10.1109/ICIP.2006.312998.

[76] J. Sung, S. Lee, and D. Kim. A real-time facial expression recognition using the STAAM. In *Proc. 18th International Conference on Pattern Recognition ICPR 2006*, volume 1, pages 275–278, 2006. doi: 10.1109/ICPR.2006.158.

[77] J. Sung and D. Kim. STAAM: Fitting a 2D+3D AAM to stereo images. In *Proc. IEEE International Conference on Image Processing*, pages 2781–2784, 2006. doi: 10.1109/ICIP.2006.313124.

[78] J. Sung and D. Kim. Real-time facial expression recognition using STAAM and layered GDA classifier. *International Journal of Image and Vision Computing*, 2008.

[79] F. Romeiro and T. Zickler. Model-based stereo with occlusions. In *Analysis and Modeling of Faces and Gestures*, pages 31–45, 2007.

[80] F. Dornaika and A. Sappa. Improving appearance-based 3D face tracking using sparse stereo data. *International Conference on Computer Vision Theory and Applications, VISAPP*, 2006.

[81] J. Liebelt, J. Xiao, and J. Yang. Robust AAM fitting by fusion of images and disparity data. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2483–2490, 2006. doi: 10.1109/CVPR.2006.255.

[82] P. Mittrapiyanuruk, G. N. DeSouza, and A. C. Kak. Calculating the 3D-pose of rigid-objects using active appearance models. In *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, volume 5, pages 5147–5152 Vol.5, 2004. doi: 10.1109/ROBOT.2004.1302534.

[83] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.

[84] R. Yang and Z.Y. Zhang. Model-based head pose tracking with stereovision. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, FGR'02*, page 255, 2002.

[85] J. Tu, T. Huang, Y. Xiong, T. Rose, and F. Quek. Calibrating head pose estimation in videos for meeting room event analysis. In *Proc. IEEE International Conference on Image Processing*, pages 3193–3196, 2006. doi: 10.1109/ICIP.2006.313066.

[86] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1964.

[87] Y. Aidarous, S. Le Gallou, A. Sattar, and R. Seguier. Face alignment using active appearance model optimized by simplex. In *VISAPP 2007: Proceedings of the Second International Conference on Computer Vision Theory and Applications*, pages 231–236, 2007.

[88] Y. Aidarous, S. Le Gallou, and R. Seguier. Simplex optimisation initialized by gaussian mixture for active appearance models. In *Proc. 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pages 79–84, 3–5 Dec. 2007. doi: 10.1109/DICTA.2007.4426779.

[89] D. Cristinacce and T. F. Cootes. Facial feature detection and tracking with automatic template selection. In *Proc. 7th International Conference on Automatic Face and Gesture Recognition FGR 2006*, pages 429–434, 2006. doi: 10.1109/FGR.2006.50.

[90] D. E. Goldberg. *Genetic algorithms in search optimization and machine learning*. Addison-Wesley, Reading, 1989.

[91] C. McIntosh and G. Hamarneh. Genetic algorithm driven statistically deformed models for medical image segmentation. *ACM Workshop on Medical Applications of Genetic and Evolutionary Computation Workshop*, 2006.

[92] P. Ghosh and M. Mitchell. Segmentation of medical images using a genetic algorithm. *8th annual conference on Genetic and evolutionary computation*, pages 1171–1178, 2006.

[93] A. Hill, T.F. Cootes, and C.J. Taylor. A generic system for image interpretation using flexible templates. In *British Machine Vision Conference*, pages xx–yy, 1992.

[94] S. Pigeon. M2VTS: Multi modal verification for teleservices and security applications, 1996.

[95] www.vision.caltech.edu/bouguetj/calib_doc/index.html. camera calibration toolbox.

[96] I. Rechenberg. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* PhD thesis, Berlin Technical University, 1971.

[97] John Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan, 1975.

[98] A. Vicini and D. Quagliarella. Multipoint transonic airfoil design by means of a multiobjective genetic algorithm. In *35th Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, 1997.

[99] S. Russel and P. Norvig. *Artificial Intelligence: A modern Approach.* Prentice Hall, 1995.

[100] H. P. Schwefel. *Evolution and Optimum Seeking.* John Wiley & Sons, 1995.

[101] W. Smart and M. Zhang. Applying online gradient descent search to genetic programming for object recognition. *Second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 133–138, 2004.

[102] E. Fernandez, M. Grana, and J.R. Cabello. An instantaneous memetic algorithm for illumination correction. *Congress on Evolutionary Computation*, 1:1105– 1110, 2004.

[103] M. Alker, S. Frantz, K. Rohr, and H. S. Stiehl. Hybrid optimization for 3D landmark extraction: Genetic algorithms and conjugate gradient method. *Bildverarbeitung fur die Medizin*, pages 314–317, 2002.

[104] P. A. N. Bosman and D. Thierens. Exploiting gradient information in continuous iterated density estimation evolutionary algorithms. *13th Belgium–Netherlands Artificial Intelligence Conference BNAIC'01*, pages 69–76, 2001.

[105] B.T. Skinner, H.T. Nguyen, and D.K. Liu. Hybrid optimisation method using PGA and SQP algorithm. *IEEE Symposium on Foundations of Computational Intelligence*, pages 73 – 80, 2007.

[106] R. Zhang and S. Ma. An efficient hybrid genetic algorithm for continuous optimization problems. *Scientia Magna*, 3:46–53, 2007.

[107] M. Zdansky and J. Pozivil. Combination genetic/tabu search algorithm for hybrid flowshops optimization. *Conference on Scientific Computing Algorithms*, pages 230–236, 2002.

[108] A. Osyczka. *Design Optimization*, chapter Multicriteria optimization for engineering design, pages 193–227. Academic Press, 1985.

[109] V. Pareto. *Cours d'Economie Politique*. 1896.

[110] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *International Conference in Genetic Algorithms*, pages 416–423, 1993.

[111] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 82–87, 1994.

[112] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, pages 221–248, 1994.

[113] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Caltech Concurrent Computation Program 826, California Institute of Technology, Pasadena, CA, 1989.

[114] M. Lahanas, D. Baltas, and N. Zamboglou. A hybrid evolutionary algorithm for multi-objective anatomy-based dose optimization in high-dose-rate brachytherapy. *Physics in Medicine and Biology*, 48:399–415, 2003.

[115] G. Li, G. Song, Y. Wu, J. Zhang, and Q. Wang. A multi-objective hybrid genetic based optimization for external beam radiation. *27th Annual International Conference of the Engineering in Medicine and Biology Society*, pages 7734 – 7737, 2006.

[116] E.F. Wanner, F.G. Guimaraes, R.H.C. Takahashi, D.A. Lowther, and J.A. Ramirez. Multiobjective memetic algorithms with quadratic approximation-based local search for expensive optimization in electromagnetics. *IEEE Transactions on Magnetics*, 44:1126 – 1129, 2008.

[117] P. A. N. Bosman and E. D. de Jong. Exploiting gradient information in numerical multi–objective evolutionary optimization. *Conference on Genetic and evolutionary computation*, pages 755–762, 2005.

[118] S. Hiwa, T. Hiroyasu, and M. Miki. Hybrid optimization using DIRECT, GA, and SQP for global exploration. *Congress on Evolutionary Computation*, pages 1709–1716, 2007.

[119] S.Z. Martinez and C.A.C. Coello. A proposal to hybridize multi-objective evolutionary algorithms with non-gradient mathematical programming techniques. *10th international conference on Parallel Problem Solving from Nature*, pages 837–846, 2008.

[120] A.R. Yildiz, N. Öztürk, N. Kaya, and F. Öztürk. Hybrid multi-objective shape design optimization using taguchi's method and genetic algorithm. *Structural and Multidisciplinary Optimization*, 34:317–332, 2007.

[121] A. Sattar, N. Stoiber, R. Séguier, and G. Breton. Gamer's facial cloning for online interactive games. *International Journal of Computer Games Technology*, 2009.

[122] K. Zuiderveld. Contrast limited adaptive histogram equalization. Graphics Gems IV edition, 1994. p.474-485.

List of Figures

List of Tables

VU :

Le Directeur de Thèse

Résumé

L'équipe SCEE de Supélec travaille dans le domaine de la radio logicielle et intelligente, encore appelée Radio Cognitive (CR - Cognitive Radio). Dans cette thèse, nous avons présenté une solution pour l'analyse de visage temps réel dans un équipement de radio cognitive. Dans ce cadre particulier, nous proposons des solutions d'analyse de visage, à savoir «l'estimation de la pose et des caractéristiques faciale d'un visage inconnu orienté ».

Nous proposons deux systèmes d'alignement de visages.

1) Le premier exploite un AAM 2.5D et une seule caméra. La phase d'optimisation de cet AAM est hybride: elle mixe un algorithme génétique et une descente de gradient. Notre contribution tient dans l'opérateur de descente de gradient qui travaille de concert avec l'opérateur classique de mutation : de cette manière sa présence ne pénalise pas la vitesse d'exécution du système.

2) Le second met en œuvre un AAM 2.5D mais exploite plusieurs caméras. La recherche de la meilleure solution découle également d'une approche hybride qui mixe une optimisation multi-objectifs : le NSGA-II, avec une descente de gradient. Notre contribution tient dans la proposition d'une méthode efficace pour extraire des informations concernant la pertinence de chacune des vues, ces informations sont ensuite exploitées par la descente de gradient.

Des comparaisons quantitatives et qualitatives avec d'autres approches mono et multi-objectifs montrent l'intérêt de notre méthode lorsqu'il s'agit d'évaluer la pose et les traits caractéristiques d'un visage inconnu.

Mots clés: Analyse de visage, Model Actif d'Apparence 3D, système multi caméras, l'optimisation multi-objectifs.

Abstract

In this study we are interested in the pose estimation and precise localization of face features such as the eyes, the nose and the mouth of an out-of-plane rotated unknown face. Main application of this thesis work is in the Cognitive Radio equipments. We place ourselves within the framework of a low quality acquisition with camera(s) installed on Cognitive Radio equipments e.g. mobile phone, laptop, desktop computers etc. The face pose and localization of facial features in an unconstrained environment are the major problems for CR equipment. All of its subsequent face related applications (e.g. face recognition, face synthesis, face data compression etc.) highly depend upon the methods used for the facial analysis system.

In order to extract face features, we use the Active Appearance Models (AAM), deformable models allowing shape and texture to be jointly synthesized. We initially propose a new 2.5D AAM, based on 3D model, which makes it possible to perform pose estimation and features localization of an oriented face.

Secondly, we propose a new optimization methodology for the face search by AAM, in a single camera system, by the hybridization of deterministic and direct search method which has never been used and tested before.

Our method hybridizes Gradient Descent (GD) inside the Genetic Algorithm (GA) in a unique way. Along with other operators of GA we propose gradient operator which works in conjunction with the mutation operator of GA thus it does not make the system computationally expensive.

Finally for a complete facial analysis system by multiple cameras, we proposed a new concept of multi-objective AAM. In this method, facial images from multiple cameras are analyzed simultaneously by 2.5D AAM. For the face search optimization we propose a unique way of hybridizing GD with NSGA-II (Non-dominating Search Genetic Algorithm-II).

Both of our propositions are robust, real time, efficient and extract facial features even in unknown and out-of-plane rotated faces.

Keywords: Face Analysis, 3D Active Appearance Model, Multi-Camera System, Multi-Objective Optimization.