Improved Facial Feature Fitting for Model Based Coding and Animation

Po Tsun, Paul, Kuo

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
June 2006

Abstract

Model-based representation of human faces has wide application in video communication, Virtual-Reality (VR) environments, Human Computer Interface (HCI) systems and facial expression studies. For example, model-based coding leads to a very low bandwidth requirement for image transmission as the images are transmitted using a small number of parameters rather than image pixels.

This thesis addresses accurate facial feature fitting and examples of applying this to model-based facial coding are given. Facial features are fitted by novel approaches based on dynamic curve fitting techniques, namely Active Contour Models (Snakes) and Deformable Templates. Active contour models are used for fitting the lips, eyebrows, and chin, while the eyes are fitted by deformable templates as the shapes of the eye and iris can be approximated to parabolas and a circle. To achieve a high success rate for model fitting in real head-and-shoulders images, a number of original techniques are described. These include: applying an adaptive colour model to the snakes for segmenting the lip and skin; developing a faster and more accurate eye template updating scheme with a novel method for eye corner detection; preprocessing the eyebrow image with lighting balancing and shadow removal; using a topologically adaptive snake for chin fitting and recovering a poorly-lit chin section by extrapolation.

The wireframe face model, Candide-3 used in this work has to be precisely adapted to the face in terms of the vertex positions, so that the texture information and global shape and animation parameters can be extracted. An approach using a texture measure combined with a measure of the fitted features to the model vertices is shown to give a promising result, demonstrating that automatic adaptation is possible.

The adapted face model is rotated to different views and animated with several expressions in order to assess the quality of the model fitting. The techniques of facial feature fitting and face model adaptation are also demonstrated for head-and-shoulders images and video sequences.

Acknowledgements

I would like to express my deepest gratitude to all who have contributed to the completion of this thesis.

First and foremost, I would like to thank my supervisors, Dr John Hannah and Dr David Renshaw, for the guidance and invaluable suggestions they provided continuously during the period of my PhD program.

Second, I would like to thank Dr Peter Hillman, for giving me access to his research results and software that my research can be built on.

Third, I would like to thank my PG/RA/RF colleagues for their great ideas, opinions, advice regarding the progress of my research.

Fourth, I would like to acknowledge the use of the XM2VTSDB and METT database and associated documentation in order to test my algorithms.

Last but not least, I would like to thank my family members, dad and mum, and my girlfriend, Helen, for their encouragements and care over the years.

Contents

List of figures

List of tables

Acronyms and abbreviations

PCA	Principal Component Analysis [15]
AAM	Active Appearance Model [23]
XM2VTSDB	Extended M2VTS Database
DFT	Discrete Fourier Transform
RGB	Red-Green-Blue colour space
CMY	Cyan-Magenta-Yellow colour space
HSI	Hue-Saturation-Intensity colour space
PCs	Principal Components
GVF	Gradient Vector Flow [34]
HSV	Hue-Saturation-Value colour space
SGLD	Space Gray-Level Dependence matrix [53]
ASM	Active Shape Models [23]
SOM	Self-Organisation Map [54]
SVM	Support Vector Machines
GA	Genetic Algorithms
HMM	Hidden Markov Model
SACM	Sampled Active Contour Model
HCI	Human Computer Interface
IR	Infra-Red
SCACM	Statistical Constraint Active Contour Model
WFM	Wire-Frame Model
MPEG	Moving Picture Experts Group
FFPs	Facial Feature Points
FAPs	Facial Animation Parameters
FACS	Facial Action Coding System
VRML	Virtual Reality Modelling Language
AU	Animation Unit

Nomenclature

$I(x, y)$	Image intensity (at position (x, y))
$\boldsymbol{E}$	an Edge field
G_x	horizontal component of an edge
G_y	vertical component of an edge
$T1$	starting threshold of the Canny edge detector
$T2$	finishing threshold of the Canny edge detector
R, G, B	normalised Red, Green and Blue values ($[1, 0]$)
H,S,I	normalised Hue, Saturation and Intensity values (S and $I \in [1, 0]$, $H \in [0, 2\pi]$)
$\boldsymbol{A'}$	the transpose of matrix $\boldsymbol{A}$
$\boldsymbol{\alpha_k}$	the k-th largest eigenvector of a covariance matrix
$\overline{\boldsymbol{x}}$	mean shape vector in AAM (Active Appearance Model)
$\boldsymbol{P_s}$	a set of orthogonal modes of shape variation in AAM
$\boldsymbol{b_s}$	a vector of shape parameters in AAM
$\overline{\boldsymbol{g}}$	mean normalised greylevel of texture vector in AAM
$\boldsymbol{P_g}$	a set of orthogonal modes of intensity variation in AAM
$\boldsymbol{b_g}$	a vector of texture parameters in AAM
$\boldsymbol{W_s}$	a diagonal matrix of weights for $\boldsymbol{b_s}$ in AAM
$\boldsymbol{Q}$	a set of orthogonal modes of appearance variation in AAM
$\boldsymbol{c}$	a vector of appearance parameters in AAM
$\boldsymbol{T_s}$	shape transform in AAM
$\boldsymbol{T_g}$	texture transform in AAM
$v(s)$	a parametrised snake (Active Contour Model)
α, β, γ, ρ	tension, rigidity, image and pressure (weighting) parameters of a snake
$v'(s)$, $v''(s)$	first and second derivatives of the snake $v(s)$ with respect to s
E_{image}	snake image Energy
E_{con}	snake constraint Energy
∇	a vector gradient on a 2D (image) domain
h	the spatial step of the snake
ϵ	colour/intensity difference for generating the pressure force of the snake

τ, σ, k	mean, standard deviation (s.d.) and colour tolerance of a colour snake
$\boldsymbol{S}$	Candide shape units
$\boldsymbol{A}$	Candide animation units
$\boldsymbol{\sigma}$	Candide shape parameters
$\boldsymbol{\alpha}$	Candide animation parameters
$\boldsymbol{R}, s, \boldsymbol{t}$	Candide global pose parameters; rotation, scaling and translation
$\overline{\boldsymbol{g}}$	the standard Candide model
$Thr_{upper\ lip}$	the upper lip (hue) threshold of the lip snake
$Thr_{lower\ lip}$	the lower lip (hue) threshold of the lip snake
F_{pressure}	snake pressure force
F_{image}	snake image force
$W_{upper\ lip}$	weighting factor applied to $Thr_{upper\ lip}$
$W_{lower\ lip}$	weighting factor applied to $Thr_{lower\ lip}$
$\phi(x, y)$	the edge field used in eye deformable templates
$R_{expected}$	the expected radius of iris used in eye deformable templates
A_{cir}, L_{cir}	the area and circumference of the deformable circle in iris fitting
W_I, W_E, W_S	weights for intensity, edge and saturation terms in iris fitting
R_{deform}	radius of the deformable circle in iris fitting
P	a variable weight to encourage lower Saturation in eye corner finding
E^+	the convolution result using the positive gradient template in eye corner finding
E^-	the convolution result using the negative gradient template in eye corner finding
W_C	the weight associated with the corner template term
a	the parameter controlling the degree of curvature of the parabola in eyelid fitting
(x_o, y_o)	the origin of the predefined axes of the parabola in eyelid fitting
(x_m, y_m)	the horizontal and vertical translation of the parabola in eyelid fitting
θ	the global rotation of the parabola in eyelid fitting
D	the distance to the parabola origin from a parabola point in eyelid fitting
$L_{par}, L^o_{par}, L^i_{par}$	the length of the parabola and the parabolic section outside and inside the iris
A_w	the white area of the eye in eyelid fitting
ϕ_o, ϕ_i	the edge field outside and inside the iris
υ	Valley term in eyelid fitting
ω	White term in eyelid fitting
$SF(x)$	scaling factor for eyebrow lighting balancing
$OC(x)$	offset constant for eyebrow lighting balancing

N_s	nominal skin intensity
N_{eb}	nominal eyebrow intensity
N_{range}	nominal skin-eyebrow range; $N_{range} = N_s - N_{eb}$
δ	the weight associated with direction force of the eyebrow snake
$v(s) = (r(s), \theta(s))$	Adaptive1D radial snake for chin fitting
$r(s)$	distance from the mouth centre to the snake control point $v(s)$
$\theta(s)$	the angle of the inclination of $v(s)$
$\vec{n}$	a normal unit vector pointing outward
μ_I	mean intensity
S_n	snake control points on the chin contour
V_n	Candide-3 vertex points
$r_x, r_y, r_z, s, t_x, t_y$	Candide global pose parameters -x, y, z rotation, scale and x, y translation
$\boldsymbol{p}$	a set of Candide parameters, including global pose parameters and shape units ($\boldsymbol{\sigma}$)
$\boldsymbol{S}$	PCA training set of face images
$\overline{\boldsymbol{s}}$	standard Candide shape
$\boldsymbol{j(i,p)}$	a normalised face shaped from a face image $\boldsymbol{i}$ using parameter $\boldsymbol{p}$
$\overline{\boldsymbol{x}}$	mean appearance of the faces in the training set
$\boldsymbol{x(i,p)}$	the reconstructed face image from $\boldsymbol{j(i,p)}$ using PCA
$\boldsymbol{r(i,p)}$	residual image; the difference between $\boldsymbol{j(i,p)}$ and $\boldsymbol{x(i,p)}$
$e(\boldsymbol{p})$	appearance error measure; $e(\boldsymbol{p}) = \|\|\boldsymbol{r(i,p)}\|\|$
$\boldsymbol{p'}$	the best-fit parameters of Candide model to the face
$\boldsymbol{U}$	an update matrix for the Candide model
$\boldsymbol{F}$	a matrix whose rows are the co-ordinates of the found feature points
$\boldsymbol{V_p}$	a matrix whose rows are positions of the corresponding Candide-3 vertices
$\epsilon(\boldsymbol{p})$	feature error measure; $\epsilon(\boldsymbol{p}) = \|\|\boldsymbol{V_p} - \boldsymbol{F}\|\|^2$
λ	a weighting factor associated with the feature error measure

Chapter 1
Introduction

1.1 Background

This thesis deals with processing a face image for model based coding and animation, topics include facial feature extraction, face model fitting and synthesis of facial expressions. Human faces are the main subject to be studied in this thesis. The face is one of the most important tools for making communication between human beings. The shape of the face (facial traits) reveals age, gender, ethnic origin and health of a person [1] while facial expressions provide measures for emotion, intention, cognition and personality [2].

To be able to identify either facial traits or expressions, knowing the locations of facial features is essential as the traits and expressions reflect the distribution of the features in the face. A model based representation is one method to represent the feature locations by using sets of more meaningful parameters. The basic idea of model based representation and coding for a face is described as follows: a source image containing a face is analysed, using image processing techniques, the face and facial features are identified. A general or specific model, which is usually a wireframe describing the 3D shape of the object, is then adapted to the face (by matching the features and face boundary, etc.). The adaptation is performed by adjusting the model's parameters, which typically are size, position and shape, and the non-rigid parameters if the face contains significant expressions. This representation of the face which allows translating a face state to sets of parameters is called the "model based representation" which differentiates from the "appearance based representation" that sees the face as a collection of pixels of varying intensity.

This representation enables highly efficient coding and transmission of the image. Instead of transmitting the full image pixel-by-pixel, or by coefficients describing the waveform of the image, the parameters extracted from the model can be coded and transmitted (model based coding). This usually compresses data significantly (typically 720 $\times$ 576 pixels down to a handful of parameters) and is beneficial to very low bit-rate applications such as video-phone and teleconferencing. To achieve acceptable visual similarity to the original image, the texture

from the original object is also transmitted. However, it is normally sent once to the receiver at the beginning of the connection so that the transmission of the model parameters is not affected.

Another use of the model based representation of the face is to generate facial expressions. This can be either artificial synthesis or a clone of expressions. Synthesising an expression artificially uses a set of rules developed by psychologists such as Ekman [3]. On the other hand, a clone of expressions requires extraction of the model parameters from one face image faithfully and transferring the parameters to animate another.

1.2 Definition of Problems

Many challenges had to be tackled during this work. The face and estimated facial feature positions were firstly located using Hillman's technique [4], which applies skin detection and PCA (Principal Component Analysis) [5]. It is insufficient having only estimated facial feature positions for the model based representation of the face. It is necessary to fit the contours of the features so that important characteristics, such as corners of eyes, corners of lips and jaw width, etc., can be extracted. These problems must be tackled in an *automatic*, *robust* and *timely* fashion. A measure of *accuracy* of fitting these contours is also required. However, such measurement can only be made subjectively as the ground truth comparison is usually unavailable. For the model fitting of the face, the information extracted from the fitted feature contours combined with the facial texture is used. Again, automation, robustness and time is the major concern. The accuracy of the final model fitting is assessed by animation. This is performed by using the model based representation so that the model parameters can be adjusted. Again, it is hard to measure animation quantitatively, so a subjective assessment has been undertaken.

1.3 Contributions of Research

The main contribution of this thesis is to develop new or extend existing algorithms for facial feature contour fitting and face model fitting. The facial features of interest are lip, eyes, eyebrows, and chin. The contours of these facial features are fitted by novel approaches based on dynamic curve fitting techniques, namely Active Contour Models (Snakes) [6] and Deformable Templates [7]. Active contour models are used for fitting the lips, eyebrows, and chin, while

the eyes are fitted by deformable templates. Face model fitting is performed by using the found feature contours in conjunction with the face texture. All of the proposed algorithms are automatic and considered robust and accurate. The time (speed) has not yet achieved a satisfactory level and this will be addressed in the conclusion chapter.

The lip fitting algorithm applies a novel colour adaptive snake. This snake is capable of automatically sampling lip and skin colour for each individual and adjusting its internal parameters accordingly. Thus the snake can extract the lip robustly regardless of age, gender, ethnicity and lighting conditions. A separate algorithm for locating lip corners is developed to improve the accuracy of the lip corner fitting of the snake [8].

The eye fitting algorithm is based on Yuille's deformable templates [7]. Several terms are added or modified in the energy function and a novel method for eye corner detection (using colour properties) is incorporated to improve the robustness and accuracy of the fitting. A new template updating scheme is developed, which now is simpler and more efficient: allowing one feature to be fitted properly before the next feature is fitted [9].

The eyebrow fitting algorithm combines a number of new techniques. A novel lighting balancing is first applied to a sub-image containing eyebrows. This is then processed by a k-means clustering and 3 clusters representing skin, dense eyebrow and sparse eyebrow are shown in the image. A wrongly clustered area which is caused by heavy shadow is removed by a novel shadow removing technique. Eventually, a twin snake algorithm is applied in the image to fit the upper and lower boundaries of the eyebrow. The intercept points of the two snakes are the corners of the eyebrow [10].

The chin fitting algorithm applies a novel topologically adaptive snake. This snake is able to "probe" the area of the image where the snake is going to develop and two statistical measures about the intensity topology of this area, mean and standard deviation (s.d.), are computed. Thus the internal parameters of the snake are adjusted according to these two statistical measures. In consequence, the chin fitting is robust against various skin types. Also a novel polar coordinate system is adopted for snake development in the chin region in order to find the chin characteristics. To increase the robustness against variation of the illumination, a new method which extrapolates the fitted chin section and uses it as the snake re-initialisation to recover the chin section in the shadow is developed [11].

The face model fitting algorithm applies a combined method of the Active Appearance Model

(AAM) [12] and a feature-based approach. A number of important feature points are extracted from the found feature contours and incorporated in an Analysis by Synthesis routine [13]. Thus now fitting the face model requires to minimise both the appearance and feature errors. This combined method significantly improves the accuracy of the model fitting at the feature points and is robust to various types of faces with moderate head rotation [10, 14].

1.4 Structure of the Thesis

The rest of the thesis is organised as follows:

Chapter 2 serves as a background study. It is divided into two main parts. The first part reviews the basic face image processing techniques such as Principal Component Analysis (PCA) [15], Active Appearance Models (AAM) [12], Active Contour Models [6] and Deformable Templates [7]. It also includes a brief review of digital (colour) image processing (edge detection, smoothing, colour space and colour transform.)

The second part of chapter 2 surveys the topics related to the work in the literature. This includes face detection, lip fitting, eye and eyebrow fitting, chin fitting and model face fitting. Various techniques are investigated and compared, and their pros and cons are addressed. Chapter 2 finishes off by introducing the image database XM2VTSDB [16] (used for testing the algorithms) and the face models [17].

Chapter 3 describes the lip fitting algorithms in detail. An initial snake fitting using a static colour model is introduced first and followed by an improved snake algorithm using an adaptive colour model. The results show the latter approach is robust against various lighting conditions.

Chapter 4 describes the eye and eyebrow fitting algorithms in detail. The first part of chapter 4 covers the eye fitting which is able to extract the iris and eyelid contours. A novel approach for detecting eye corners is also included. The second part of the chapter 4 concentrates on eyebrow fitting, which is a cascaded technique consisting of lighting balancing, k-means clustering, inner corner shadow removal and twin snake eyebrow extraction.

Chapter 5 describes the chin fitting algorithms in detail. The chin is fitted by a novel topologically adaptive snake and an approach to recover the chin under the shadow is also introduced.

Chapter 6 describes the model face fitting in detail. Two model fitting techniques are presented

in this chapter. The first one relocates the model vertices according to the extracted feature positions after performing an appearance-based model fitting. The better fitting technique (the second technique in this chapter) is to incorporate a feature measure into the appearance-based fitting algorithm. Thus the model is fitted when both the feature and appearance errors are minimised. Both techniques show that the fitting of the model at the feature points is significantly improved .

Chapter 7 draws the conclusions of this thesis. It also points out the limitations of the algorithms and suggests possible future research.

Chapter 2

Background Study and Literature Review

2.1 Review of Major Image Processing Techniques

There are a number of essential image theories and processing techniques which are frequently mentioned and applied in this thesis. It is important to understand them before moving onto advanced methods and new ideas.

2.1.1 Basic Theory: Digital Image, Edge Detection, Smoothing, Colour Spaces and Colour Transform

Digital Image

A digital image is an image composed of a set of dots or picture elements (pixels), the smallest units of the image. Each pixel is assigned a discrete numerical value to represent the intensity (brightness), thus forming a greyscale digital image. The smallest numerical value corresponds to the lowest intensity, which is black. On the other hand, the largest numerical value gives the highest intensity (white). The dynamic range of a digital image counts the number of numerical values available for a pixel. For example, an 8-bit greyscale image indicates that each pixel of the image has 256 shades from the lowest (black) to highest intensity (white). Modern digital images usually contain colour. A digital colour image is a digital image that includes colour information for each pixel. For visually acceptable results, it is necessary to provide three colour layers (channels) for each pixel. Each layer is similar to a greyscale image - providing shades of the colour. The blend of the three layers results in the colour of the pixel, which can be interpreted as coordinates in some colour space. The colour space and transformation is discussed later in this section.

Edge Detection

Edge Detection is used to enhance the appearance of edges. The edge, by definition, is the high frequency part of the image. The word "frequency" here is used to describe "change in space domain" rather than a more familiar description of "change in time domain". In other words, an edge ($\boldsymbol{E}$) in a 2D (2 Dimensional) image is given in the following form:

$$\boldsymbol{E} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix} \tag{2.1}$$

$$|\boldsymbol{E}| = \sqrt{G_x^2 + G_y^2} \tag{2.2}$$

where $I(x, y)$ in Equation 2.1 is the intensity of a pixel at position (x, y) in a Cartesian grid of the image. It is a custom in the image processing community that x coordinate is incremented rightwards and y coordinate is incremented downwards, i.e., the origin $(0, 0)$ is at the top left corner of an image. G_x and G_y are horizontal and vertical components of the edge ($\boldsymbol{E}$). Equation 2.2 then shows the magnitude of $\boldsymbol{E}$.

¿From Equation 2.1 and 2.2, many discrete forms of the edge detection were derived. Some researchers are working on discrete frequency domain (i.e. through a Discrete Fourier Transform (DFT)) [18] to obtain more quantified edge detection. However, this approach generally suffers from slow processing in DFT. Thus many researchers (and the author) prefer more straightforward approaches of using spatial filters.

Consider an image sub-region shown as a 3×3 matrix in Figure 2.1. By intuition, the magnitude of the edge is given by $\sqrt{(I_5 - I_8)^2 + (I_5 - I_6)^2}$ or $\sqrt{(I_5 - I_9)^2 + (I_6 - I_8)^2}$. For convenience, they can be approximated by $(|I_5 - I_8| + |I_5 - I_6|)$ or $(|I_5 - I_9| + |I_6 - I_8|)$. The latter one is implemented by taking the absolute value of the convolution response of the two 2×2 spatial filters (or masks) shown in Figure 2.2 and summing the results. These masks are called the *Roberts cross-gradient operators* [18]. Sometimes masks of even sizes are awkward to implement. It leads to another approximation $|(I_7 + I_8 + I_9) - (I_1 + I_2 + I_3)| + |(I_3 + I_6 + I_9) - (I_1 + I_4 + I_7)|$. The difference between the third and first row of the 3×3 region approximates the derivatives in the y-direction, and the difference between the third and first column approximates the derivative in the x-direction. This can be realised by a set of spatial filters called the *Prewitt operator* [18], as shown in Figure 2.3. Furthermore, the coefficients

near the centre of the mask can have higher values to reflect the importance of proximity. This results in a *Sobel edge detector* (Figure 2.4). It is not unusual to use only one rather than the pair of the filters to solely enhance the edge in either the vertical or horizontal direction.

$$\begin{bmatrix} I_1 & I_2 & I_3 \\ I_4 & I_5 & I_6 \\ I_7 & I_8 & I_9 \end{bmatrix}$$

Figure 2.1: *An image sub-region*

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Figure 2.2: *Roberts operator*

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 2.3: *Prewitt operator*

A more complicated (but better) edge detection can be done by the *Canny edge detector* [18]. The *Canny operator* works in a multi-stage process. First of all the image is smoothed by a *Gaussian smoothing filter* (see next section). Then a 2D edge operator (such as Roberts, Prewitt or Sobel) is applied to the smoothed image to highlight the edges. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as *non-maximal suppression* [18]. The tracking process exhibits hysteresis controlled by two thresholds: $T1$ and $T2$ with $T1 > T2$. Tracking can only begin at a point on a ridge higher than $T1$. Tracking then continues in both directions out from that point until the height of the ridge falls below $T2$. The major advantages of the Canny edge detector are that the detected edges form thin lines instead of scattered points and the user has control of its parameters such as the width of the Gaussian operator, $T1$, and $T2$. The implementation of a Canny edge detector is documented in [19].

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 2.4: *Sobel operator*

Smoothing

Smoothing filters are used for blurring and for noise reduction. Blurring is used for removal of small details from an image prior to (large) object extraction or for bridging of small gaps in lines or curves [18]. Noise is reduced by taking average/median over the centre pixel's and its neighbours' intensities.

The most common and simplest smoothing filter is an *average filter* [18]. It is a $N \times N$ matrix with all its entries equal to one and is attached by a factor of $\frac{1}{(N \times N)}$, where N is very often an odd number. The matrix in Figure 2.5 illustrates a 3×3 average mask. A bigger N offers more powerful blurring ability. To reflect the importance of proximity, a *Gaussian smoothing filter* is used. It contains Gaussian coefficients in its matrix which uses larger values for entries closer to the centre. Figure 2.6 shows a 5×5 Gaussian smoothing filter with the width = 1 s.d. (standard deviation).

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 2.5: 3×3 *average mask*

$$\frac{1}{273} \times \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Figure 2.6: 5×5 *Gaussian smoothing filter*

In some cases when the objective is to reduce the noise rather than blurring, a median filter is used. This method is particularly effective when the noise pattern consists of strong, spike like components and the characteristic to be preserved is edge sharpness. A 3×3 median filter is applied to the region and the process is to replace the intensity of the centre pixel by the median of the 9 sorted samples.

Colour Space and Colour Transform

The fundamental colour space used in image processing is RGB space (Red, Green, Blue space). It is a 3D space constructed from the so-called *primary colours* - Red, Green, and Blue which represent as x-, y- and z- axes of the space (see Figure 2.7). In this space, any colour is produced by the combination of different proportion of Red, Green and Blue and all the colours are found as points on or inside the colour cube. Notice that the greyscale is shown as a diagonal line connecting Black at the origin to White at point $(1, 1, 1)$ [1].

Although RGB space is an easy way to model colour, it has several drawbacks. First, it is difficult to alter the intensity as change in an individual colour channel will not only alter the intensity, but also alter the resultant colour. Second, RGB representation is not consistent with human perception. Psychologists know human eyes are more sensitive to one primary colour than another and therefore the space should be skewed somehow. For these reasons, there are many other colour spaces developed, such as CIELAB, CMY, YIQ and HSI, to suit particular applications. Among these, HSI is the most frequently used colour space in this thesis apart from RGB.

HSI is short for *Hue, Saturation* and *Intensity*. *Hue* is a colour attribute that describes a pure colour (e.g. pure yellow, orange, or red), whereas *Saturation* gives a measure of the degree to which a pure colour is diluted by white light (lower value means more diluted). *Intensity* again represents the brightness of the colour. The HSI colour space owes its usefulness to two principal facts. First, the intensity now is decoupled from the colour information in the image, thus the intensity can be altered without changing the resultant colour. Second, the hue and saturation components are intimately related to the way in which human beings perceive colour.

Figure 2.8 shows HSI space as a "double cone" model. Hue (H) represents the colour as an

[1] the normalised values $[0, 1]$ are used in this colour cube

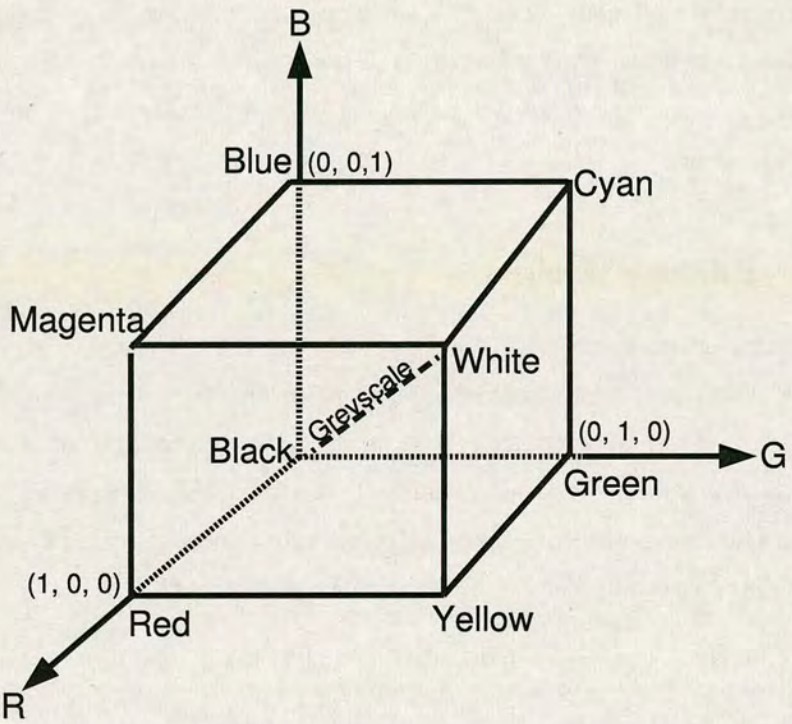


Figure 2.7: *RGB colour cube. The diagonal line connecting the Black (0, 0, 0) and White (1, 1, 1) is the greyscale.*

angle, varying from 0^o to 360^o. The three primary colours divide the plane equally, Red is at 0^o, Green is at 120^o and Blue is at 240^o. Saturation (S) corresponds to the radius, varying from 0 to 1. Intensity (I) varies along the vertical axis with 0 being black and 1 being white. When S = 0, the colour is a grey value of intensity and H is undefined. When S = 1, the colour is fully saturated (i.e. pure colour). The greater the saturation, the farther the colour is from white/grey/black (depending on the intensity). When I = 0, the colour is black and therefore H is undefined again. By adjusting I, a colour can be made darker or lighter. By maintaining S = 1 and adjusting I, shades of that colour are created.

The conversion between RGB and HSI spaces is not straightforward but is well documented in [18]. Here the equations for conversion are given.

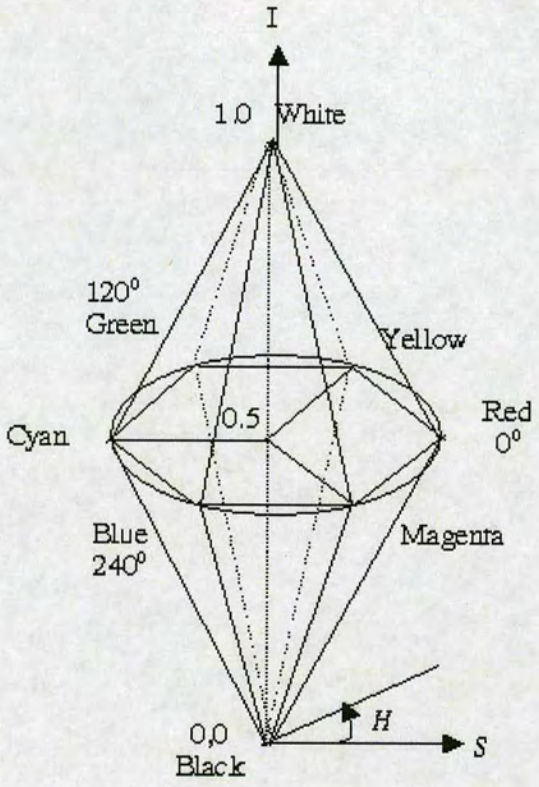


Figure 2.8: *HSI colour space [20]*

For converting RGB to HSI:

$$
\begin{aligned}
I &= \frac{1}{3}(R+G+B) \\
S &= 1 - \frac{3}{(R+G+B)}[\min(\mathrm{R},\mathrm{G},\mathrm{B})] \\
H &= \cos^{-1}\{\frac{\frac{1}{2}[(\mathrm{R}-\mathrm{G})+(\mathrm{R}-\mathrm{B})]}{[(\mathrm{R}-\mathrm{G})^2+(\mathrm{R}-\mathrm{B})(\mathrm{G}-\mathrm{B})]^{\frac{1}{2}}}\}
\end{aligned}
$$

$$
\text{when} \qquad B > G,\ H = 2\pi - H \tag{2.3}
$$

where R,G, B, I and S are in the range [0,1]. H is in radian from $0 \sim 2\pi$.

For converting HSI to RGB, 3 scenarios according to the angle of Hue are considered:

$$
\begin{aligned}
\text{when} \quad & 0 < H \leq \tfrac{2}{3}\pi \\
b &= \frac{1}{3}(1 - S) \\
r &= \frac{1}{3}[1 + \frac{S\cos(\mathrm{H})}{\cos(\frac{\pi}{3} - \mathrm{H})}] \\
g &= 1 - (r + b)
\end{aligned} \tag{2.4}
$$

$$
\begin{aligned}
\text{when} \quad & \tfrac{2}{3}\pi < H \leq \tfrac{4}{3}\pi \\
H &= H - \frac{2}{3}\pi \\
r &= \frac{1}{3}(1 - S) \\
g &= \frac{1}{3}[1 + \frac{S\cos(\mathrm{H})}{\cos(\frac{\pi}{3} - \mathrm{H})}] \\
b &= 1 - (r + g)
\end{aligned} \tag{2.5}
$$

$$
\begin{aligned}
\text{when} \quad & \tfrac{4}{3}\pi < H \leq 2\pi \\
H &= H - \frac{4}{3}\pi \\
g &= \frac{1}{3}(1 - S) \\
b &= \frac{1}{3}[1 + \frac{S\cos(\mathrm{H})}{\cos(\frac{\pi}{3} - \mathrm{H})}] \\
r &= 1 - (g + b)
\end{aligned} \tag{2.6}
$$

In Equation 2.4, 2.5 and 2.6, r, g and b are the normalised version of R, G and B.

$$
\begin{aligned}
r &= \frac{R}{(R+G+B)} \\
g &= \frac{G}{(R+G+B)} \\
b &= \frac{B}{(R+G+B)}
\end{aligned} \tag{2.7}
$$

2.1.2 Principal Component Analysis (PCA), Eigenfaces and Active Appearance Models (AAMs)

Principal Component Analysis (PCA)

PCA has become a popular technique for object recognition in image processing. The history of PCA is long. The earliest version of PCA was dated back to 1901, proposed by Pearson [15]. Originally, PCA was used in applied statistical mathematics. *"The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming it to a new set of variables, the Principal Components (PCs), which are uncorrelated, and which are ordered so that the first* **few** *retain most of the variation present in* **all** *of the original variables"* [15]. The mathematical form of PCA transformation is written as:

$$
\boldsymbol{z} = \boldsymbol{A'x} \tag{2.8}
$$

where vector $\boldsymbol{x}$ is the original data set to be transformed by PCA and vector $\boldsymbol{z}$ is the new data set obtained by PCA transformation of $\boldsymbol{x}$.

$\boldsymbol{A'}$ is the transpose of matrix $\boldsymbol{A}$, whose k-th column, $(\boldsymbol{\alpha_k})$, is the k-th eigenvector of the covariance matrix of data set $\boldsymbol{x}$. If the eigenvectors are arranged in such a way that the k-th eigenvector corresponds to the k-th largest eigenvalue of the covariance matrix, it turns out that vector $\boldsymbol{z}$ will have the first number corresponding to the greatest variation and the last number corresponding to the smallest variation. The full derivation of PCA can be found in [15].

Why is the PCA so useful in image processing? Imagine that an image (called the matching image) is compared with a pool of images to find the most similar one. The simplest way is to compare any images in the pool with this matching image pixel by pixel (assuming all the images were scaled to the same size). If every pixel in an image can be seen as a dimension of

variation, an image with the size of $M \times N$ pixels will consist of $M \times N$ dimensions.

Therefore all the dimensions are required to be compared across all the images in the pool. It is obvious that it is highly computational expensive and the error in each dimension has no indication of order of importance. PCA is very useful in solving this. The principal components (PCs) are firstly computed by taking several images in the pool. Images are then compared in a lower space constructed by using only a number of most important PCs instead of using all dimensions. It speeds up the matching process and makes sure that all the major dimensions of variation are included. One concern of PCA is that the images need to be "normalised" before operating PCA. This includes adjusting all the images to the same size and to a common orientation. A better performance can be obtained if histogram equalisation[2] and DC component removal are undertaken prior to PCA.

Eigenfaces

One relevant example to this PhD work is Turk and Pentland's approach [5, 21] to the detection and recognition of human faces by applying PCA. Their approach transforms face images into a small set of characteristic feature images, called "eigenfaces", which are the principal components of the initial training set of face images. These eigenfaces emphasise the variation of the global "features" of the faces which may or may not be directly related to human intuitive notion of face features such as the eyes, nose, lip and hair (examples of eigenfaces are shown in Figure 2.9). Then a sub-space ("face space") is built with a number of the "best" eigenfaces (corresponding to the largest eigenvalues) to include the most dominant facial variations. Recognition is performed by projecting a new image into this face space and then classifying the face by comparing its position in face space with the positions of known individuals. The summary of their face identification and recognition process is given as follows:

- Initialisation: Acquire the training set of face images and calculate the eigenfaces, which define the "face space".

- When a new face image is encountered, calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto the "face space".

[2]histogram equalisation redistributes the intensity distribution of the image so that the dynamic range and contrast of the pixels are increased [18]

Figure 2.9: *Images show the most important eight eigenfaces corresponding to the largest eigenvalues [22]*

- Recognition: Determine if the image is a face at all (whether known or unknown) by checking to see if the image is sufficiently close to the training face clusters.

- Identification: If it is a face, classify the weight pattern as either a known person or as unknown.

Their approach takes advantage of PCA for obtaining and comparing the faces in a much reduced dimension space. Furthermore, they also provide a fast way to compute the eigenvectors and eigenvalues [5], calculated in the order of the number of training images (normally < 50) rather than the order of the number of pixels in images (typically 720×576). To recognise faces with broader orientation, it is advisable to include faces with different orientations (e.g. a "frontal" view, a "45^o" view and a "profile" view) in the training set.

Active Appearance Models (AAMs)

While Turk and Pentland's approach [5, 21] projects only the face texture in the lower dimension, Active Appearance Models (AAMs) [12, 23–25] represent both the shape and texture variations seen in a training set of images. The training set consists of labelled images, where key landmark points (such as eye, eyebrow, and lip corners) are marked on each example object. In consequence two statistical models, the shape model and texture model, are built by using a PCA dimensionality reduction method. Given a face image as an example, a shape model capable of representing the plausible faces is given below.

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{P_s b_s} \tag{2.9}$$

Where $\overline{\boldsymbol{x}}$ is the mean face shape vector which contains the labelled points at eye, eyebrow, lip corners and so on. $\boldsymbol{P_s}$ is a set of orthogonal modes of shape variation computed by PCA, and $\boldsymbol{b_s}$ is a vector of shape parameters.

To build a texture model for plausible faces, each example face image is warped so that its control points match the mean shape (using a triangulation algorithm [12]). Then the intensity information from the *shape-normalised* image over the region covered by the mean shape is sampled. Again, PCA is applied on the normalised data to obtain the texture modes.

$$\boldsymbol{g} = \overline{\boldsymbol{g}} + \boldsymbol{P_g b_g} \tag{2.10}$$

Where $\overline{\boldsymbol{g}}$ is the mean normalised greylevel vector, $\boldsymbol{P_g}$ is a set of orthogonal modes of intensity variation and $\boldsymbol{b_g}$ is a set of texture parameters

To reduce complexity as well as take advantage of the correlation between the shape and texture variations. The $\boldsymbol{b_s}$ and $\boldsymbol{b_g}$ are further combined.

$$\begin{bmatrix} \boldsymbol{W_s b_s} \\ \boldsymbol{b_g} \end{bmatrix} = \boldsymbol{b} = \begin{bmatrix} \boldsymbol{Q_s} \\ \boldsymbol{Q_g} \end{bmatrix} \boldsymbol{c} = \boldsymbol{Q c} \tag{2.11}$$

Where $\boldsymbol{W_s}$ is a diagonal matrix of weights to raise $\boldsymbol{b_s}$ to the similar scale to $\boldsymbol{b_g}$. $\boldsymbol{Q}$ is a set of orthogonal modes and $\boldsymbol{c}$ is a vector of *appearance* parameters controlling both the shape and

texture of the model.

Now an appearance model for faces has been constructed. The vector $\boldsymbol{c}$ in Equation 2.11 controls both the shape and texture models (i.e. an appearance model), as shown in Equation 2.12 and Equation 2.13.

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{P_s W_s^{-1} Q_s c} \tag{2.12}$$

$$\boldsymbol{g} = \overline{\boldsymbol{g}} + \boldsymbol{P_g Q_g c} \tag{2.13}$$

To match the appearance model to a face in the image (i.e. reconstruction of a face image). Two transform functions $\boldsymbol{T_s}$ and $\boldsymbol{T_g}$ are required. "Shape Transform" ($\boldsymbol{T_s}$), including scale, translation and rotation parameters, transforms the shape $\boldsymbol{x}$ to the desired shape. "Texture transform" $\boldsymbol{T_g}$, including intensity scaling and offset, transforms the texture $\boldsymbol{g}$ to the desired texture. The procedure of the reconstruction applies an "Analysis by Synthesis" routine [13, 26]. It starts with estimating the model parameters $\boldsymbol{c}$ and $\boldsymbol{T_s}$. Thus a shape of the image patch can be located and pixels in this region are sampled and projected back to the texture model frame. This image projected texture is compared with the texture composed by $\boldsymbol{c}$ using Equation 2.13. The difference is called the "residual image" and an iterative framework described in [12, 24–26] is employed to minimise it.

2.1.3 Active Contour Models (Snakes)

Active Contour Models (Snakes) are dynamic curves for extracting deformable objects and are commonly applied for face and facial feature extraction [27–31]. The snake was first proposed by Kass et al. [6]. It is a curve guided by internal and constraint forces, and influenced by image forces (often termed external forces) that pull it toward features such as lines and edges. Mathematically, the snake is a parametric curve (denote as $v(s)$ in Equation 2.14), that moves around in the image domain to minimise the energy function regarding internal (including tension and rigidity), constraint and image (external) forces (Equation 2.16).

$$v(s) = [x(s), y(s)], s = [0, 1] \tag{2.14}$$

$$E_{snake} = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))ds \quad (2.15)$$

$$E_{snake} = \int_0^1 \frac{\alpha}{2}|v'(s)|^2 + \frac{\beta}{2}|v''(s)|^2 + E_{image}(v(s)) + E_{con}(v(s))ds \quad (2.16)$$

Where α and β are weighting parameters that control the snake's tension and rigidity, respectively. $v'(s)$ and $v''(s)$ denote the first and second derivatives of $v(s)$ with respect to s. The external energy function E_{image} is derived from the image so that it takes on its smaller values at the features of interest, such as boundaries. E_{con} represents the optional constraint term which may be neglected for simplicity.

The Euler-Lagrange equation (Equation 2.17; E_{con} has been dropped out) can be used to find the minimum point of Equation 2.16. It differentiates each term in Equation 2.16 and turns the "energy minimisation" problem to a "force balancing" problem.

$$-(\frac{\alpha}{2}v')' + (\frac{\beta}{2}v'')'' + \nabla E_{image}(v) = 0 \quad (2.17)$$

If the feature of interest is an intensity edge, $E_{image}(v)$ can be rewritten with:

$$E_{image}(v) = -\gamma|\nabla I(v)|^2 \quad (2.18)$$

$I(v)$ denotes the intensity of the image. ∇ represents the vector gradient on the 2D (image) domain. Thus the curve is attracted by the minimum of the potential, which means the local maximum of the gradient, which is an intensity edge. γ is the weighting factor associated to this term.

To obtain a numerical solution for Equation 2.17, a finite differences approach is employed.

$$\begin{aligned} F(v) = {} & \frac{1}{h}(a_i(v_i - v_{i-1}) - a_{i+1}(v_{i+1} - v_i)) + \\ & \frac{1}{h^2}(b_{i-1}(v_{i-2} - 2v_{i-1} + v_i) - 2b_i(v_{i-1} - 2v_i + v_{i+1}) + b_{i+1}(v_i - 2v_{i+1} + v_{i+2})) - \\ & \gamma\nabla|\nabla I(v_i)|^2 \end{aligned} \quad (2.19)$$

where h is the spatial step and v_i, a_i and b_i are the discrete quantities of the snake parameters. That is, $v_i = v(ih)$, $a_i = \frac{\alpha(ih)}{h}$ and $b_i = \frac{\beta(ih)}{h}$. $F(v)$ is the "resultant force" which pushes the snake closer to the feature at a temporal step in the iteration. When $F(v)$ is reduced to zero, it turns out that there is no force and the snake is stationary, indicating an energy minimum (and possibly the feature of interest) is reached.

A downside of this original snake is that the snake must be initialised close to the feature of interest, as further away from the feature produces additional local minima to which the snake could be falsely attracted. The simplest way to fix this problem is to produce a blurred version of the image, hence the image gradients of the feature can be spread into a wider area to overcome the local minima. However, this blurring approach still has some drawbacks; (a) the image gradients can not be extended indefinitely by blurring. In fact, the image gradients can only be spread to overcome the local minima within a few pixels surrounding it. (b) some less strong edges may be suppressed due to smoothing causing the snake to be unable to converge.

Toward this end, many new active contour models were developed for (at least partially) solving this initialisation problem.

Balloon Snake

For closed-loop snakes, Cohen et al. [32, 33] suggested that an extra term of "pressure force" should be added into the snake equation of force, thus Equation 2.17 becomes Equation 2.20. The pressure force makes the snake act like a balloon, which can inflate or deflate depending upon the direction of the pressure force. The $\left(\frac{\partial v}{\partial s}\right)^{\perp}$ denotes the direction of the normal at a position on the snake and ρ is a weighting factor (a positive/negative value indicates an inflating/deflating force). In consequence, the initial snake would not be necessary at the proximity of the feature of interest. What the initial snake requires is to place itself inside or outside the object of interest and the appropriate direction of pressure force is applied. This framework can also be used in open-loop snakes provided that the initial direction for snakes to evolve is known.

$$-(\frac{\alpha}{2}v')' + (\frac{\beta}{2}v'')'' + \nabla E_{image}(v) - \rho\left(\frac{\partial v}{\partial s}\right)^{\perp} = 0 \tag{2.20}$$

Gradient Vector Flow (GVF) Snake

Xu and Prince [34] proposed a different approach to solve the initialisation problem and a related problem that snakes are unable to coverage into concave shapes. They introduced a dense vector field called Gradient Vector Flow (GVF) field, which is the replacement for the original external force of the snake. The GVF field is derived from the greylevel of the edge map of the image by applying the "*diffusion theory*". Thus the resultant field has a large capture range allowing the snake to "feel" the edge (the feature) before getting near it. GVF snake is more computational expensive than the balloon snake but it is advantageous to use GVF snake when the snake is required to converge to concave shapes or the initial convergent direction is unavailable.

Colour Snake

Colour snakes do not attempt to solve the initialisation problem. Instead, it improves the extraction results by making use of the fact that recent images are acquired and processed in colour. Generally speaking, colour snakes have the following advantages: (1) snakes can be attracted to the object with desired colour. (2) desensitising the effect of change in lighting (e.g., the shadow) which cause false edges, hence enabling snakes to extract the tangible edges, rather than illusive ones.

Two typical colour snake models are introduced; Seo et al. [27] proposed using dual-colour patches in the snake's control points [3]. The new control points are shown in Figure 2.10. Each control point has outer and inner colour patches, which are specified by the user for purpose of colour matching. Thus, the snake now is not only subject to minimising the energy function (Equation 2.16), but also subject to colour matching on both sides of the snake curve. For instance, a snake for lip detection will have pink-red colour on the inner patch and skin colour on the outer patch. Hence, a snake finds the lip when it reaches an edge as well as the inner and outer colour are both matched. Nevertheless, this algorithm works well only when the desired colours are exactly known in advance. If the colours are unknown, Seo et al. [35] provided a heuristic approach which is colour adaptive, but it is rather computational complex and its performance appears to be heavily dependent on the choice of weights.

[3] the discrete form of the snake is represented by points forming the curve or loop. These points, moved under the influence of internal and external forces of the snake, are called snake's "control points".

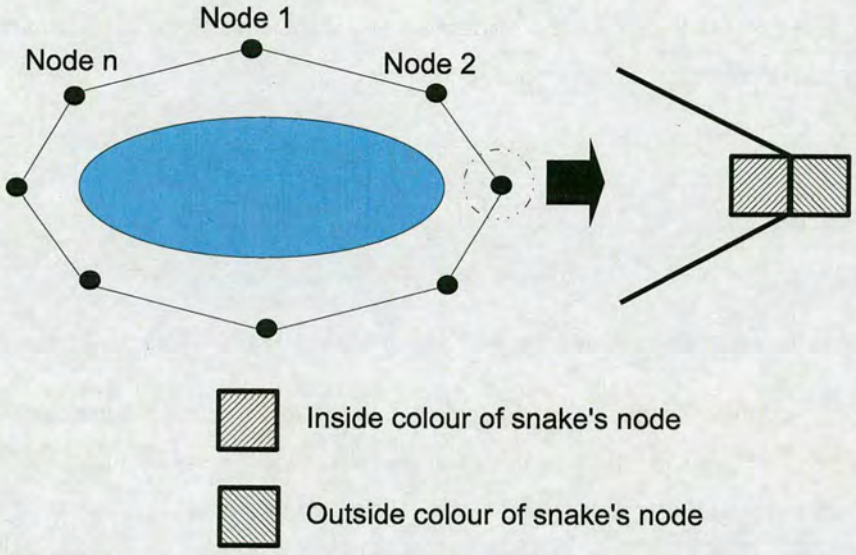


Figure 2.10: *Colour snake model and its dual-colour control points [27].*

Schaub and Smith [36] developed a colour snake model allowing the desirable colour with certain variation. First of all, a statistical colour model was built in Hue-Saturation-Value (HSV) colour space [4]. The mean and standard deviation (s.d.) of the desirable colour were evaluated in Hue, Saturation and Value channels, respectively. Next, they combined Cohen's balloon snake [32] with the Gaussian colour model. The pressure term was incorporated with the colour distribution and is expressed as $\rho \left(\frac{\partial v}{\partial s}\right)^{\perp} (\epsilon_{pressure} - 1)$ (this was $\rho \left(\frac{\partial v}{\partial s}\right)^{\perp}$ in Equation 2.20 previously), where $\epsilon_{pressure}$ was defined in Equation 2.21. This time the weighting factor (ρ) is fixed to a positive number and the magnitude and direction of the pressure force are controlled by $(\epsilon_{pressure} - 1)$. In Equation 2.21, τ_n, σn and k_n $(n \in \{1, 2, 3\})$ are the mean, standard deviation (s.d.) and colour tolerance in Hue, Saturation and Value channels, respectively. p_1, p_2 and p_3 denote pixel's intensity in each H, S, and V channel.

k_n sets the acceptable margin deviated from the mean (τ_n) in terms of s.d. (σ_n) in each channel. For instance if k_n is set to a small value (say 0.001 s.d.), then the colour of the pixel has to be very close to the mean colour to allow this pixel being extracted. (i.e. to make the pressure force = 0). If k_n is set to a very large value (say infinite), then $\epsilon_{pressure} \approx 0$, making the pressure force = $1 \times \rho$ and the snake will expand to the image boundary (i.e. accepting all the pixels). So, by varying the value of k_n, users have control of the colour margin, ranging from exact

[4] HSV is a similar colour space to the aforementioned HSI but using slightly different formulation.

colour matching to accepting all colours in the spectrum. In their work, Schaub and Smith [36] set k_3 to an infinity so the object was extracted solely depending on the chromaticity (i.e. Hue and Saturation), regardless of the intensity (i.e. Value).

$$\epsilon_{pressure} = \sqrt{\left(\frac{p_1 - \tau_1}{k_1\sigma 1}\right)^2 + \left(\frac{p_2 - \tau_2}{k_2\sigma 2}\right)^2 + \left(\frac{p_3 - \tau_3}{k_3\sigma 3}\right)^2} \tag{2.21}$$

2.1.4 Static Template Matching and Deformable Templates

The static template matching algorithms are used to locate objects in the image which do not undergo much deformation. In face detection and recognition, this technique is often used to extract ears and noses. Normally a template is constructed artificially (e.g. sketched edge maps [37]) or by sampling the feature in the sequence (e.g. [38, 39] created a mean feature template from samples of the features). Then the correlation is carried out between the template and the search areas in the image. The place having highest correlation value is most likely to be the location of the feature. To extend the usability of the static templates, many researchers allow the templates having a range of size variations so that the same feature with different scales can be extracted from the image.

However, static template matching cannot cope with deformable objects as the template is fixed in shape. To tackle this drawback, Yuille et al. [7] developed the technique of "Deformable Templates" to extract deformable objects. As its name indicates, deformable templates adjust their own shapes in "some degrees" when matching is carried out. Normally the template is chosen from the primary shapes (e.g., circle, parabola, ellipse, etc.) and the adjustment of the shape is by updating the parameters in its shape's mathematical description (e.g., radius, curvature, focal points, etc.). A cost function is associated with each template and is used to measure the fitness of the template to the feature of interest. By updating the parameters and moving the templates across the search area, the cost function can be maximised/ minimised which indicates that the feature is fitted.

As it only updates the parameters of the shape rather than changes the shape type, this implies that the deformation of the template is limited in the same shape family. For instance, a deformation of a parabola will still be a parabola (but might be with different curvature and in different orientation). The advantage of using a deformable template is not only that the feature can be located and fitted but also that a set of shape parameters can be identified. Thus the repro-

duction of the feature in the latter stage is easy. To extend the usability of deformable templates, several deformable templates can be concatenated to form a compound one [37]. However the cost function of this compound template is sometimes difficult to optimise. Highly deformable features which can not be approximated by the primary shapes are usually extracted by Active Contour Models (Snakes) instead. In the next section, examples of using deformable templates to fit chins, lips and eyes will be addressed.

2.2 Literature Survey

2.2.1 Previous Work on Face Detection

Detecting faces in images is the very first step towards any facial image processing. However producing a robust and accurate face detection routine is still a challenging task as the face is nonrigid and has a high degree of variability in size, shape, colour, and texture. In this section, numerous techniques for face detection and recognition are reviewed. This is followed by a description of the selected algorithm developed by Hillman et. al [4], which produced benchmark points on the facial features for this work. Using the definition in [40] for face detection: given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face.

Broadly speaking, face detection can be categorised into 4 groups according to the methods used. They are:

1. Knowledge-based methods. These rule-based methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the geometric relationships between facial features.

2. Feature-based approaches. These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to find faces.

3. Template matching methods. Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed to determine the existence of the faces in the input image.

4. Appearance-based methods. In contrast to template matching, the models (or templates)

are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection.

In the first group, the face detection methods are developed based on the rules derived from the researcher's knowledge of a human face. The rules are drawn to describe the features of a face and their geometric relationships. For example, a face often appears in an image with two eyes that are symmetric to each other with a nose and a mouth on this symmetry axis. The geometric relationships between features can be represented by their distance, positions and size. Facial features in an input image are searched first and face candidates are identified based on the coded rules.

The downside of this approach is the difficulty in converting human knowledge into well-defined rules. If the rules are set to be strict, they may fail to detect faces that do not pass all the rules. On the other hand, if the rules are too general, they may detect non-face objects as faces. Moreover, a set of rules usually can only be applied for certain poses of faces (e.g. frontal view, 45^o side view, or 90^o profile view). The number of rules has to be enormously big to exhaustively cover all possible cases. Yang et al. [41] used a hierarchical knowledge-based method to detect the face. In their algorithm three levels of rules were used; the first level rules, telling what a face looks like, work on the coarse version of the image while the last level rules, relying on details of facial features, are applied to the highest resolution of the image. The number of candidate faces in three levels are in descending order. This coarse-to-fine framework reduces a great deal of computation as only a few faces enter the last level which is most computationally expensive.

This idea of multi-resolution hierarchy is frequently applied [42]. Kotropoulos and Pitas [42] presented a rule-based face detection similar to [41] by locating the facial features using Kanade's technique [43]. The boundary and features of candidate faces are located by "Horizontal and Vertical projection" [43]. Subsequently, a set of detection rules (which are hierarchical) are applied to the found features to validate the presence of the face.

The second group of approaches (feature-based) aims to find invariant "features" of face for detection. The underlying assumption is based on the observation that humans can effortlessly detect faces and objects in different poses and lighting conditions and so, there must exist properties or features which are invariant over these variabilities. Common features used for detection include facial features [44, 45], textures [46], skin colour [47, 48] and combination of

features (integration of facial features, textures, skin colour, size, and shape) [49].

Leung et.al. [44] developed a probabilistic method to detect a face in a cluttered scene based on local feature detectors and random graph matching. They formulated the face detection problem as a search problem in which the goal is to find the arrangement of certain facial features that is most likely to be a face pattern. To this end, five features (two eyes, two nostrils and a lip) are extracted and their relative distance is computed over an ensemble of images and modelled by a Gaussian distribution.

In [45, 50, 51], Yow and Cipolla presented a feature-based technique that used implicit facial features. These implicit features were extracted by a second derivative Gaussian filtering and grouped to form feature regions. Measurement of a region's characteristics, such as edge strength and intensity variance are computed and stored in a feature vector. From the training data of facial features, the mean and covariance matrix of each facial feature vector are computed. A face is detected in an input image if the *Mahalanobis* distance[5] between the corresponding feature vectors is below a threshold.

Augusteijn and Skufca developed a method that inferred the presence of a face through the identification of face-like textures [52]. The textures are computed using second-order statistical feature, including "angular second moment", "contrast", "correlation", "inverse different moment", "entropy" and "sum average", on sub-images of 16×16 pixels and calculated "Space Gray-Level Dependence matrix" (SGLD) [53]. Three types of features - skin, hair and others - were used to form "face texture" and classified by a Kohonen self-organising feature map [54]. Furthermore, Dai and Nakano [46] also applied SGLD model with extra colour information incorporated with face texture model.

Human skin colour has proven to be an effective feature for face detection, as many studies have shown that the major colour difference between human skin is in intensity rather than chrominance [47, 55, 56]. The common approach is to define a skin colour region in an appropriate colour space by setting up several thresholds. A patch is classified as a skin if the pixel values fall within the colour region and vice versa. Researchers have used colour spaces such as RGB [57, 58], Normalised RGB [59–61], HSV/HSI [49, 62–64], YCrCb [65, 66], YIQ [46] and CIE [67, 68].

[5] given two vectors $\tilde{x}$ and $\tilde{y}$ with the same dimension, Mahalanobis distance between them is $((\tilde{x}-\tilde{y})' C^{-1}(\tilde{x}-\tilde{y}))^{\frac{1}{2}}$. Where C is the covariance matrix of $\tilde{x}$ and $\tilde{y}$.

Various researchers suggested that by using a combination of features, a face can be detected more convincingly. For example in [49, 63, 69], face candidates can be obtained by skin texture or colour detection followed by a connected component analysis. They are then verified by shape constraints (a face should be of elliptic or oval shape) and relationships of the local features (eyes, eyebrows, nose and mouth).

A general difficulty of this group of approaches is to define the discrimination thresholds. As these features (such as second-order statistical features, colour, texture and so on) are abstract, the decision boundaries which can separate faces and non-faces usually are not straightforward and require further computation (for example, using neural networks or PCA (Principal Component Analysis) to obtain a linear space).

Regarding the face detection by template matching (the third group of approaches), Sakai et al. [70] attempted to match the faces by using manually created templates. They used several sub-templates for the eyes, nose, mouth and face contour to model a face. Each sub-template is defined in terms of line segments. A correlation is applied between the input image's edge map and face contour template to find face candidates. This is followed by subsequent correlations between sub-images of the input and feature sub-templates to verify the face candidates. Craw et al. [71, 72] modified Sakai's approach to become more robust by allowing different scales and orientations of the templates. Later researchers developed more "flexible" template matching by applying dynamic curve/shape fitting. For example Yuille et al. [7, 73, 74] used deformable templates for face and facial feature detection and localisation, Lam and Yan [75] and Kwon et al. [76] applied Active Contour Models (Snakes) to locate the face boundary. Cootes and Taylor [77, 78] and Lanitis et al. [79] used Active Shape Models (ASM) to locate and track faces.

Generally speaking, this group of approaches are quite effective in detecting features with distinct contours (such as face boundary). However, since only the information of the shapes and outlines is used, these approaches will see some difficulties on extracting the features dominated by their contents (such as lips) without incorporating other measures.

In contrast to the template matching methods where templates are predefined by experts, methods in the last group, the appearance-based approach, uses "appearance templates" which are learned from examples in images. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face/nonface

images. The learned characteristics are in the form of distribution models or discriminant functions that are consequently used for face detection. Meanwhile, dimensionality reduction (PCA) [5] is usually carried out for the reason of computation efficiency.

Turk and Pentland [5], as mentioned in the earlier section, applied PCA (Principal Component Analysis) on a training set of face images to generate an image subspace (called the face space) in which the face detection is performed. They projected face and nonface images to this space in the training phase and formed face and nonface clusters. To detect the presence of a face in an input image, the image is projected into the same space and a distance metric to the clusters is computed. Another way to reduce face dimensionality is to use neural networks [80, 81]. Agui et al. [80] proposed an early method using neural networks to detect the face in monochromatic images. They used a two-layer network structure with two subnetworks in each layer. In [82], Vaillant et .al. used convolutional neural networks to detect faces in images. Examples of face and nonface images of 20 $\times$ 20 pixels are first created. The first neural network is trained to select face candidates from image areas. Then these candidates are verified by the second network. Burel and Carel [83] proposed an approach adopting Kohonen's Self-Organisation Map (SOM) [54] so a large number of training examples of faces and nonfaces can be compressed into fewer examples. Recently, Support Vector Machines (SVM) [84] and other kernel methods such as Genetic Algorithms (GA) [85], Naive Bayes Classifier [86] and Hidden Markov Model (HMM) [87] have been proposed for face detection. These methods implicitly project patterns to a higher dimensional space and then form a decision boundary between the projected face and nonface patterns.

The major difficulty of this group of approaches is to construct an unbiased, normalised training set. The face images in the training set should contain all kinds of possible variations, for example, different gender, age, ethnicity, beards, glasses, expressions, and so on. Furthermore, it is important to "align" all the face images before the training procedure can be carried out. These include normalisation of the size, orientation and brightness of the face images. In summary, to ensure appearance-based approaches perform properly, a large and diverse data set, and a huge amount of manual work are required.

The face detection and facial feature localisation method used in this work was developed by Hillman et al. [4]. The technique overlaps between the aforementioned group 1, 2 and 4 approaches. At the beginning, a skin colour detection was used to find the face candidates in the image (this is a group 2 approach). The CIE-Lab colour space was selected for performing

the skin detection as Hillman et al. [4] argued that skin and non-skin discrimination works better in this colour space. A set of 70 face images from a wide range of sources are then used to find the decision boundaries in the CIE-Lab space. The distribution of the pixels sampled from these faces is shown in Figure 2.11(a). This distribution can be seen to be skewed with respect to axes, thus the conventional rectangular bounding box cannot be fitted. To solve this, they applied PCA (Principal Component Analysis) to find a new linear space.

In the linear space, the limits ($p_{max} = (x_{max}, y_{max}, z_{max})$ and $p_{min} = (x_{min}, y_{min}, z_{min})$) derived from the PCA transformed training set are used as the vertices of the bounding box for skin thresholds. The colour of the input image is then segmented as follows. Each pixel is processed in turn, converted into CIE-Lab space, and then into PCA space. If the position of the pixel (p_x, p_y, p_z) in the transformed PCA space satisfies Equation 2.22 , it is marked as skin (black), otherwise as non-skin (white) (see Figures 2.11(b) and 2.11(c)).

$$\begin{aligned} x_{min} &< p_x < x_{max} \\ y_{min} &< p_y < y_{max} \\ z_{min} &< p_z < z_{max} \end{aligned} \tag{2.22}$$

This will leave holes in eye regions as eye colour is very different from skin colour. A region growing technique was used to fill the eye regions and remove the noise. Only sufficiently big non-skin regions in the image will remain as eye candidates after the region growing. Then an eigen-eye matching which is similar to Turk and Pentland's [5] was performed on the eye candidates (a group 4 approach) and a geometrical analysis was used to verify the eye positions (a group 1 approach). All eye candidates were subtracted from a 60 $\times$ 30 mean eye block and then projected into a PCA space. A matching measure (m) was obtained as the distance between each eye candidate and the centre of the cluster of known eyes in the PCA space. The smaller m is, the higher the possibility that the eye candidate is an eye.

Since scaling, orientation and noise can affect the PCA matching, a further geometrical examination was undertaken. The following constraints were used [4]:

- The magnitude of the angle of the line joining the left and right eyes must be less than 30^o.

- The distance between the regions must be between 10 and 20 % of the image width
- Both eyes must lie within the central 50 % of the image.

The unrejected eye candidate pair with the smallest combined m was taken as the position of the eyes.

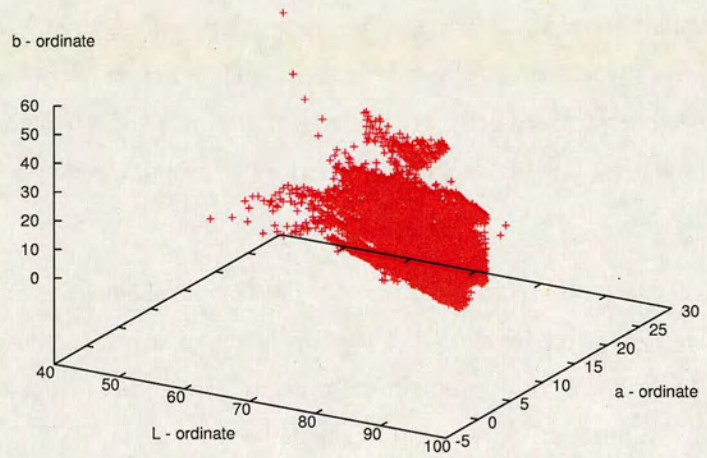
The nose and mouth were found by applying eigen-mouth and eigen-nose matching [5] in a search area defined by the found eye pair. The search region lies on the perpendicular bisector of the line connecting the found eyes. The region is 0.7 w wide and 1.5 w high, where w is the width between the eyes. These factors were found by analysing a wide range of face images [4].

Final geometrical constraints were used to remove false mouth and nose matches. The eye-nose distance (d_n) and eye-mouth distance (d_m) were computed as an average distance from the eye to the nose and from the eye to the mouth, respectively. The ratio of (d_m/dn) must be in the range (1.3, 2.6) to validate the positions of mouth and nose.

The output from this algorithm is the found face region, estimated eye, mouth and nose positions. These become the benchmarks for the feature contour fitting, which is one of the main contributions of this thesis. The entire system is depicted in Figure 2.12.

2.2.2 Previous Work on Lip Fitting

Lip fitting has been a popular research topic for computer vision scientists over past years as a wide range of applications (such as model-based coding [8, 37, 88] and lip-reading [30, 89, 90]) are based on fast and accurate lip contour fitting. Deformable Templates and Active Contour Models (Snakes) are the two most commonly used techniques in lip fitting. Yuille et al. [7, 73] were the pioneers applying the deformable templates on lip fitting. They used several conical shapes to describe the states of open and closed mouths. The cost functions of the templates include edge potential (for the upper and lower lip boundaries), valley potential (for the boundary between the upper and lower lips) and internal potentials such as potentials forcing the lip symmetrical, dipping the template at the middle of the upper lip. Their methods have obtained an initial success. However, due to high complexity of optimisation of the cost functions, several modified versions were proposed [91, 92]. Zhang [88] simplified Yuille's model by employing only 3 parabolas for closed mouth conditions and 4 parabolas for open



(a) skin points in CIE-Lab space

(b) input image

(c) skin detector

Figure 2.11: *The skin colour distribution in CIE-Lab space and the result of the skin detection [4].*

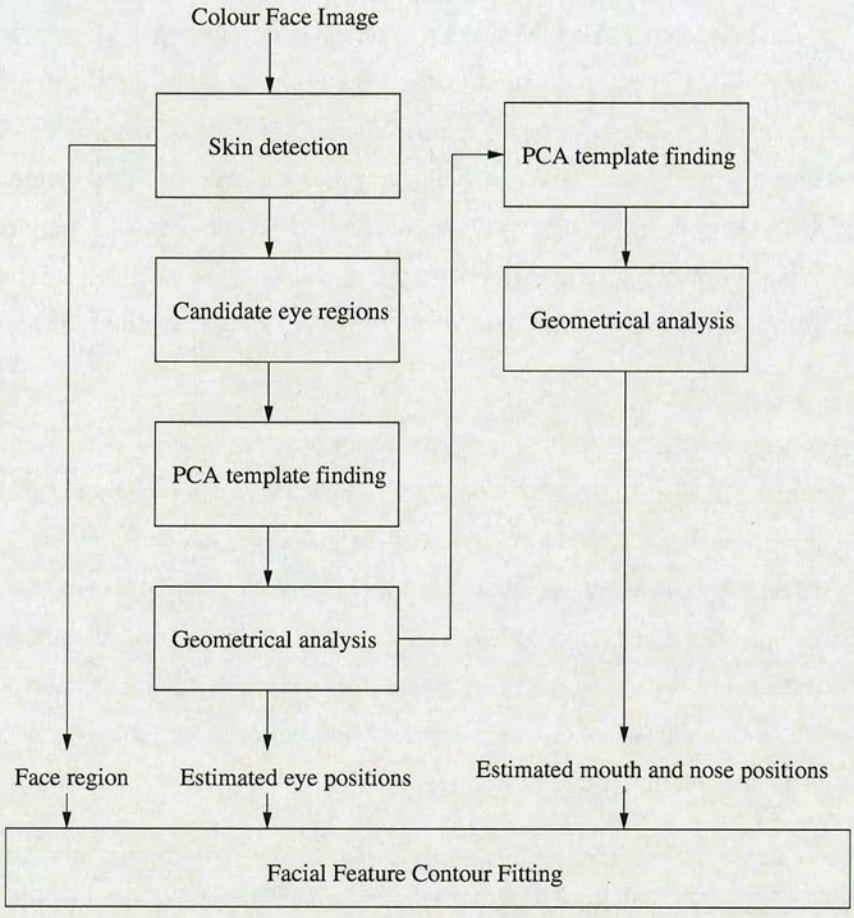


Figure 2.12: *Flowchart shows the process of estimating the face region and feature locations.*

mouth conditions with an automatic mouth state detection. The number of the parameters used by Zhang is substantially reduced to less than half of the number used by Yuille et al. Along with an improved optimisation algorithm, a fast lip fitting was achieved .

Active Contours Models (Snakes) provide higher flexibility for deformation and has proven to offer better results for the lip contour fitting [89, 90, 93, 94]. Based on Kass's snakes [6] and Cohen's balloon pressure [32], Shinchi et al. [90] and Sugahara et al. [89] developed a Sampled Active Contour Model (SACM) to fit the lip contour. Shinchi [90] employed a special repulsion force as an external force making the snake stay on the edge more robustly. Sugahara [89], on the other hand, added an extra vibration factor, helping the snake to slip through noise in the image. Eveno et al. [95] used a different approach from Cohen's to solve the initialisation issue. They developed a jumping snake which is able to grow and jump from a point quite far away from the lip. The snake grows first and hence it can "probe" the change of greyscale. In turn the snake jumps towards higher gradient of greyscale and a new snake grows where is closer to the lip edge. As a result, every iteration makes the snake approach the lip edge and eventually the lip can be extracted.

The performance of the lip fitting can be enhanced by incorporating other techniques with Active Contour Models. Barnard et al. [30] proposed a snake guided by a two-dimensional (2D) lip shape template. The external force of the snake was produced by a 2D correlation between the matched 2D template and the control points of the snake. It overcome the problem that the weak lip edge could not produce enough force to hold the snake. By applying this new external force, a good lip shape can be guaranteed in snake convergence. Okubo and Watnabe [31] tracked lips in sequences of images by applying optical flow [96] to predict the initial position of the snake in the subsequent image.

Various researchers have investigated applying colour snakes to lip fitting, as a distinguishable colour difference exists between the skin and lip regardless gender, age and ethnic origin. Moreover, a snake using colour information can avoid being attracted by fraudulent edges which are resulted from illumination differences.

As mentioned earlier, Seo et al. [27] proposed a snake with "dual-colour patches" control points. The inner patches are of lip colour while the outer patches are of skin colour. The lip boundary can be fitted by matching the colour patches with the lip and skin colours on the image (Figure 2.10).

2.2.3 Previous Work on Eye and Eyebrow Fitting

Eye Fitting

Accurate eye fitting is a key step for many computer vision applications such as model-based facial coding [4, 9, 97, 98], facial expression recognition [99–103], human computer interface (HCI) [99–101, 104–106] and biometric identification [107–109]. The term of "eye fitting" comprises finding and locating one or more of the eye features: pupil, iris, eye corners and eyelids, but excludes simply eye location estimation. Depending upon the type of applications, one or many of eye features are required to be fitted with good accuracy.

Yuille et al. [7, 73] used deformable temples to fit the holistic eye. They used two parabolas and a circle to describe an eye. Each parabola represented the upper or lower eyelid and the circle between these represented the iris. Figure 2.13 depicts the general layout of Yuille's templates.

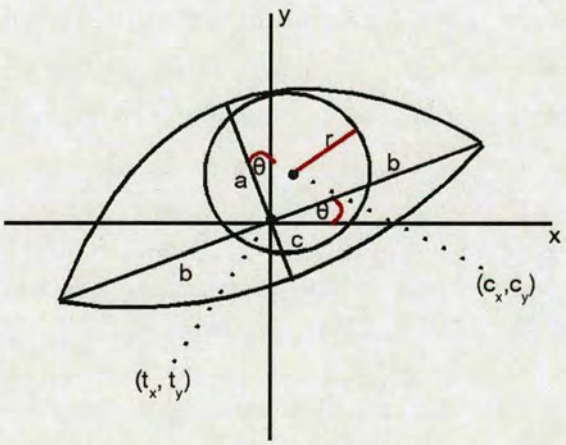


Figure 2.13: *Yuille's eye template [7].*

The cost function associated with this set of deformable templates consists of 11 parameters and 10 weights. The steepest descent algorithm combined with a cascade updating procedure was then used to minimise the cost function. It is not easy and time consuming to update such a massive set of parameters and weights, and the steepest descent approach frequently produced a result which converged to local minima. To this end, many researchers [110–112] developed better deformable eye templates and updating algorithms.

Deng and Lai [110] simplified Yuille's template and obtained a cost function consisting of only 9 parameters and 6 weighting coefficients. Instead of using an energy minimising technique to

deform the template, Deng and Lai introduced regional forces which utilised the local feature information that could not be obtained by the global descriptor. By doing this, they also avoided the cumbersome step of adjusting the weights to ad hoc values. After the major updating procedure finishes, Deng and Lai proposed a post-processing which is capable of refining the eye fitting.

Lam and Yan [111] also adopted Yuille's eye template, but they developed an extra algorithm to detect eye corners. In their approach, they found four fundamental eye corners at the beginning stage (two joints of upper and lower eyelids and two intercept points of the iris circumference and the upper eyelid). The intercept points of iris circumference and lower eyelid are less discernible due to low grey-level contrast hence they are not considered. By enforcing the parabolas to pass through these corner points, the performance of eye fitting could become faster and more accurate. However, the success rate of their eye corner finder was not reported in their contribution and their approach does not apply for widely open eyes.

Hammal et al. [103] also used deformable templates to perform the eye fitting. They argued that the upper eyelid was fitted less accurately by using a parabola as it induced a vertical symmetric constraint. Therefore they employed a Bezier curve, permitting asymmetrical bending on the curve, to fit the upper lid. They still chose the parabola and circle templates to fit the lower lid and iris as Yuille et al. did.

Apart from the deformable template approach, various researchers employed other techniques and were also capable of finding all or some eye features.

Zhu and Yang [113] proposed an approach in finding the eye corners and iris. They introduced a pair of corner operators (showing in Figure 2.14) to find the eye corners. They initially defined search regions around the eye corners and performed a two-dimensional (2D) convolution on the regions with the predefined corner operators. The positions achieving the highest result from the convolution were very likely to be the eye corners. Finding the eye corners helped to locate and fit the iris. In [113], an elliptical shape, rather than the more popular circular shape, was used for fitting the iris in a large gaze direction.

Campadielli [98] suggested a three-stage eye fitting framework. In the first stage, template matching is used in finding horizontal "eye bands" which are rectangular sub-images containing both eyes. Then neural network classification is applied on the down-sampled or wavelet-transformed eye bands to obtain the separate locations of the left and right eyes in the stage

two. In the final stage a geometrical constraint is used to justify the found eye locations from the stage two.

Perez et al. [114] proposed a method to eliminate reflection in iris region which made their iris/pupil fitting system more robust in real situations. Xie et al. [115–117] proposed an improved iris tracking algorithm allowing larger eye and head movements. They used intensity centroid to locate the centre of iris and adopted two Kalman filters to compensate the large eye and head movements.

$$\begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 2.14: *Eye corner operators [113].*

Other related work such as gaze estimation [106, 118–121] and blinking detection [102, 122] are popular in the literature. Estimation of the gaze direction is frequently applied in Human Computer Interface (HCI) systems [106, 120] as using eye gaze for replacement of the keyboard and mouse is proposed. This requires very high accuracy in tracking the movement of the pupil. The mainstream technique is to illuminate the eye by Infra-Red (IR) light and a detector is used to receive the reflection from the pupil so that the gaze direction can be inferred [120]. Beymer [121] reported that using two sets of stereo cameras plus an IR emitter can achieve sub-pixel accuracy in estimating the eye gaze. The wide angle stereo system detects the movement of the head and steers the narrow stereo system to track the pupil. Their approach outperforms others as the additional set of cameras can accommodate large head movements without intrusive head-worn gear.

Detection of eye blinking is important in eye tracking, as the interior of the eye becomes invisible when the eye closes and thus the tracking needs a special handling over this period. Tian et al. [102] proposed a dual-state model representing the open and closed eyes. When the eye is open, Yuille's template model is used. When the eye is closed, a straight line connecting the eye corners is used. The closed eye is detected if the brightness inside the iris template is higher than a predefined threshold, assuming the iris radius is known and the template is situated at the place with maximum edge response. Al-Qayedi et al. [122] emulated the eye blinking by

a model-based approach. They fit a mesh model on an open eye frame and thus the sequence of the eye blinking can be emulated by drawing the eyelid vertices closer. In operation of their eye tracking algorithm, this artificial sequence is compared to the real sequences to determine whether a blinking occurs and thus the algorithm is switched between the tracking and blinking modes. In the tracking mode, a template matching technique is used to maintain the tracked location of the eyes. In the blinking mode, an arbitrary 6 frames interval (25 frames per second) are inserted for eye blinking.

Eyebrow Fitting

The motion of eyebrows usually indicates the presence of an expression. Although the eyebrow fitting is important, few researchers have considered this in detail. Many have only extracted a few representative (3 - 5) eyebrow points [102, 123, 124] but this is inadequate for accurate face model fitting. Other researchers [37, 98, 125, 126] applied low level techniques such as intensity thresholding and edge detection to segment eyebrows. These techniques are not likely to prove effective for different types of eyebrows (colour, thickness, density of eyebrow hair) or to be robust against illumination changes. Kampmann [37] and Chen et al. [127] have paid special attention to eyebrow fitting. Kampmann reported a systematic approach using several constraints on the calculation of the eyebrow's intensity/intensity gradient and his approach was able to pick up partially occluded eyebrows. However, he only considered intensity and intensity gradient thus it is likely to have difficulty with changes of illumination. Chen [127] used k-means clustering and a snake to fit the eyebrow from a horizontally and vertically divided search window, which was more likely to obtain a continuous contour of the eyebrow. However, the illumination issue again remained unsolved.

2.2.4 Previous Work on Chin and Cheek Fitting

Fitting of the chin and cheeks has received much attention from computer vision researchers. It is a major requirement for face recognition, face modelling and facial expression analysis. However, due to its inherent complex characteristics in greyscale distribution, conventional edge detection methods have failed to extract the chin and cheeks [128]. Many researchers used geometric models to approach the boundary by dynamic curve fitting. Conventional techniques include Active Contour Models (Snakes) and Deformable Templates.

Kampmann [37, 129] proposed a deformable template approach for chin and cheek fitting. He used four parabolas, two for each, to fit the chin and cheek. He first estimated the locations of the essential facial features, such as the eyes and the mouth, and placed the chin template on an estimate initial position. Since the locations of the essential features are known, he can limit the template deformation in a smaller search space. He fitted the chin followed by cheek fitting. Both chin and cheek fitting runs with three concatenated cost functions to obtain a global optimum. He argued that his approach offered more accurate fitting results than the single parabola fitting scheme used by Xiaobo et al. [130]. However, Kampmann did not show results for his approach in an extensive set of face images.

Hu et al. [131] expressed concern that Kampmann's approach only worked on a parabola-like chin. Thus, they proposed a combined scheme of Deformable Templates and Active Contour Model. The chin contour is initially fitted by a parabola using a deformable template approach by identifying three landmark points on the chin. This initial fit is then used as the initialisation of a snake, thus a more refined chin contour can be extracted. In their snake procedure, they employed a GVF (Gradient Vector Flow) snake [34] to obtain "long range capturability", which allowed the snake to be attracted by the feature of interest from a long distance.

Sun et al. [28] used a SCACM (Statistical Constraint Active Contour Model) to fit non-parabola-like chins. The mean and variance of the face shapes were obtained in the training phase by manually marking the points on the face boundary [23]. These two statistical parameters are used as a "shape reference" in the later snake extraction phase. By doing this, they can resolve the snake problem of no geometrical sense to the feature and a much smoother face boundary (including the chin and cheeks) can be extracted. In addition, their approach allows multiple initialisation of SCACM to obtain the global optimal fit.

Goto et al. [132] attempted to fit the chin by making use of both frontal and profile view images. Thus it is easier and more accurate to find the three landmark points defining the chin - left and right extremities of the chin from the frontal image and the bottom tip of the chin from the profile image. Instead of directly employing traditional edge detection, they proposed an image processing flow to enhance the chin edge. This includes smoothing, splitting the chin areas and finding the chin edge by a number of direction vectors. The enhanced chin edge is then extracted by a snake approach. In order to improve the speed of snake evolution and keep a reasonable chin shape, they introduced symmetry and shape terms in the snake function. Their technique is fast but is unable to fit faces with large rotation.

Wang et al. [128] developed a scheme which can extract and classify chins. They first extracted a number of "chin points" near the chin contour by exploiting the intensity and edge properties. Some infeasible chin points are then removed by examining the continuity property of the points. Next they used a least square fitting technique to fit the chin points with three chin models, namely the round, pointed and trapezoidal chin models. Depending upon the minimum least square error calculated, the chin is classified as one of these three types. The data set used in their test experiment is fairly big and a numerical result is given (95.3% extraction rate and 92.0% classification rate are achieved). The result is encouraging, but difficulties in extracting the chin from complex backgrounds were also reported.

2.2.5 Previous Work on Model Face Fitting

Generating a precisely fitted head model automatically and robustly is the ultimate goal for model-based representation of human heads. A huge amount of research effort has been put in and many different techniques, depending upon applications, can be found in the literature. Some researchers used laser scanners and high resolution face models (e.g., with hundreds of thousands of surface polygons) to represent any tiny bits of the human face. In this thesis, the techniques are focused on applications using nearly frontal faces. These applications include videophones, teleconferencing, computer games and face recognition on close-to-front-view images.

In [133], Liu et. al. proposed an approach capable of finding a face geometry and orientation from face images and reconstructing the face texture from a video. Their approach aimed at low user interaction - the requirements are two nearly-frontal face images (no specified viewpoints) and a video of the turning head. They found the face geometry and orientation first and followed by facial texture reconstruction. To find the face geometry and orientation, five benchmark points corresponding to the two inner eye corners, nose tip and two mouth corners are located on these two images manually. It is then followed by an automatic face recognition- by subtracting the images (assuming the background is stationary) and exploiting colour property of skin.

Once the face is found, more corners on the face are located and their motions are estimated. In [133], a Plessey corner detector [134] is used, which defines corners as high curvature points in the intensity dimension of the face. To link found corners from one image to another a cross -correlation is applied and then a geometric constraint is used to filter the false matched corners. Finally the relative head motion (rotation and translation) between the two images can then be

estimated by applying a non-linear least-squares technique [135, 136] on the matched corner points and 5 benchmarks. The corner points and benchmarks can be reconstructed in 3D space with respect to the viewpoint used in one of the two face images. As a result, a 3D face mesh model can be fitted by minimising the distance between model vertices and found points and benchmarks in 3D space. Further improvement in local details can be achieved by finding the features using snakes [137].

Since two-view face images do not cover the whole face surface, Liu et al. [133] made use of the video sequence of the turning head to obtain a complete texture map. First they used the previous technique to fit the head model in each image in the sequence. Then a texture blending procedure similar to Pighin et al. [138] was used. The procedure is summarised below:

For each vertex on the face mesh, the blending weight is computed for each image based on the angle between surface normal and the camera direction. If the vertex is invisible, its weight is set to 0.0. The weights are then normalised so that the sum of the weights over all the images is equal 1.0. Thus a texture map composed of all the images in the video is obtained. This texture map is smoother, more realistic and has less artefacts.

In [137], Liu et al. modify the algorithm so that it can fit a 3D face model onto a single 2D frontal face image. The depths of the resulting face models are usually not accurate, but the models look recognisable in view angles not too far from the front and the fitting speed is much faster than his previous technique [133].

Luo and King proposed a fast face model fitting approach in [139]. To start with, a number of important facial features are extracted to help fitting the WFM (Wire-Frame Model) onto a 2D face image. They developed several techniques, including using neural networks and active contour models to automatically extract the facial features from the image [140]. The extracted features consist of the mouth, eyes, head boundary and mass centre of the face.

In the fitting stage, the face is fitted first with correct geometry and orientation (estimated by the found features). It is followed by facial feature alignment. Since the head boundary in the image has been extracted, the position of each boundary vertex of the WFM can be fitted precisely. This is done by stretching/contracting the boundary vertices of the WFM to match the face boundary points which constitute the same angles (with respect to the face centre) as the vertices (with respect to the WFM centre). Each inside vertex of the WFM is then adjusted in proportion to the translation of its corresponding head boundary point. Notice that since

there is no z-dimension information from a 2D image, the general WFM is adjusted in the z direction by an amount proportional to its changes in a 2D image plan, which gives a reasonable estimation.

Pighin et al. [138] developed an accurate face modelling and animation using five views of a human head, stepping from the left profile to the right profile. The manual work includes locating important feature points in different views of the images (typically, eye and mouth corners, nose tip, etc.). These points are used to automatically recover the camera parameters (position, focal length, etc.) corresponding to each photograph, as well as the 3D positions of the marked points in space. The 3D positions are then used to deform a generic 3D face mesh to fit the face. At this stage, additional corresponding points may be marked to refine the fit. The texture of the face thus can be extracted from the fitted face. They repeat the aforementioned process for the same object with several different facial expressions. Now, a database of 3D model fitted facial expressions of this person is obtained. At the animation stage, a new expression is created by interpolating between 2 or more different 3D models in the database. Meanwhile, the texture can be blended to create a more realistic look. The interpolation is not only able to be applied between holistic faces, but also allowed to apply for different regions of the face so that a "hybrid" facial expression can be created (for example, neutral expression of the upper face plus smile expression of the lower face produces a "fake smile" expression).

Blanz proposed a texture-based fitting approach [1] which is able to solve two common limitations in automated face reconstruction and animation. The first one is referred to position corresponding problem between different faces and the second one is to separate realistic faces from non-natural faces. In addition, their approach addresses a possible way to measure the face attributes. A big face dataset is collected at the onset for computing the average face and the main modes of texture variation (using PCA). A probabilistic distribution is then imposed on the morphing function to avoid unlikely faces in the reconstruction stage. This is particular useful when fitting a 3D model to a 2D image (in this case, many non-face-like solutions will be generated but removed by the constraint.). The new face is generated by linearly combining the average face and the modes with suitable weights. For finding the position correspondence of the faces, since now almost any faces can be turned into a parametric face model, the corresponding problem can be solved by a mathematical optimisation. They also derive parametric description of face attributes such as gender, age, distinctiveness, weight of a person, etc. This

is done by manually picking up faces with particular attributes in the dataset and creating a sub-dataset for them. Then PCA is applied to this sub-dataset to obtain the variation.

Cootes et al. [12, 24, 25] developed "Active Appearance Models" (AAMs) which are capable of using both shape and texture information to perform face fitting (see Section 2.1.2). The face is fitted by an "Analysis by Synthesis" routine which aims to minimise a "residual image" between the projected face image and the reconstructed one. Figure 2.15 illustrates this routine. It contains a feedback loop which can hold the analysed parameters (e.g., global pose, shape and/or animation parameters of the wire-frame model) and thus the face can be reconstructed in the synthesis block. The reconstructed face and input face are then normalised and compared in the lower dimensional space (by PCA). The error (the residual image) between these two images is computed. The optimal face fit is obtained by minimising this residual image. Ahlberg [13, 26] extended Cootes et al's work and applied it to real-time face tracking.

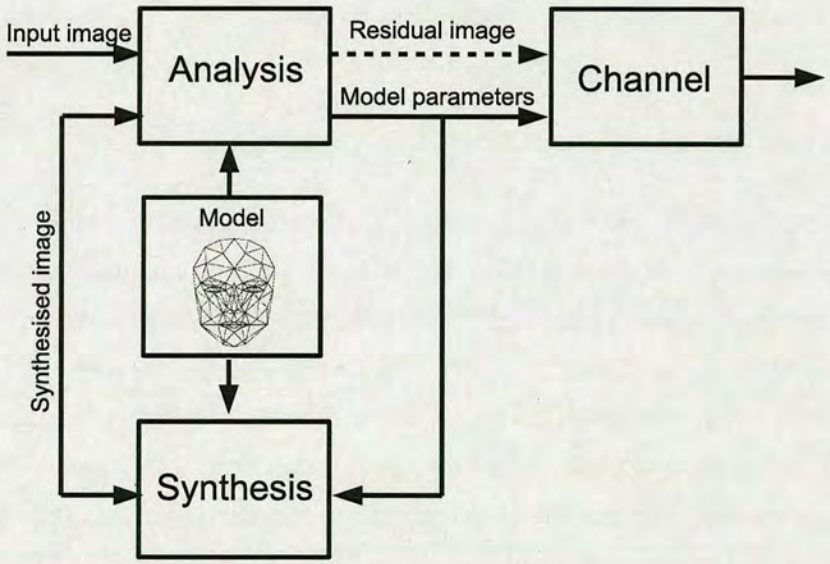


Figure 2.15: *The analysis by synthesis routine [13].*

The major problem for these model fitting techniques is the initialisation. Most of them require manual fitting of the feature points (even the appearance-based approaches also require features to be fitted so that an initial fit of the model can be estimated) at the initial step. Moreover, having only a few located feature points cannot achieve high accuracy in model fitting. Instead, feature contour fitting which gives the complete feature's boundary points is required. These

issues of the manual process and fitting accuracy have to be solved in order to apply these techniques to real applications.

2.3 Image Database - The Extended M2VTS Database (XM2VTSDB)

To assess the feature and model fitting algorithms developed in this work, a collection of images - the extended M2VTS database (XM2VTSDB) [16] from the University of Surrey - is used. This is a database of head-and-shoulders colour images with colour depth of 8-bit per channel (resulting in total 24-bit per pixel) and resolution of 720 $\times$ 576 pixels. It contains images of 295 subjects shot in several scenarios [141]. The most common one is the frontal view position which consists of 8 images for each subject showing different hair style, clothing, expressions and make-up. Other scenarios include profile views (showing either left or right side of the face) and half-lit faces (restricting illumination from either left or right). The size of the data set, the range of subjects and their characteristics (89% white, 46% female, 35% with glasses, 13% with facial hair), and the scenarios enable a realistic test for the fitting algorithms.

2.4 Wire-Frame Face Model-The Candide-3

Candide-3 is the wire-frame face model used in this work for face model fitting, face animation, and demonstration of Model-based Coding. It is chosen since (1) it is relatively simple in terms of the number of its vertices and surface polygons, allowing users to rapidly reconstruct and animate the face model, (2) it is publically available and (3) it is compatible with the standard face animation tool MPEG-4 [142–145]. Candide-3 is the successor of Candide-1 [146] and -2 [147]. All versions of Candide have the same characteristics - they contain a low number of vertices and surfaces which enable fast manipulation by users. Table 2.1 lists the number of vertices and surfaces for each Candide model and Figure 2.16 displays the wire-frame model of Candide-3.

	Candide-1	Candide-2	Candide-3
Number of vertices	79	160	113
Number of surfaces	108	238	168

Table 2.1: *List of number of vertices and surfaces for Candide-1, -2 and -3 [17].*

To adapt and animate the Candide-3 model, three sets of control elements are provided. The

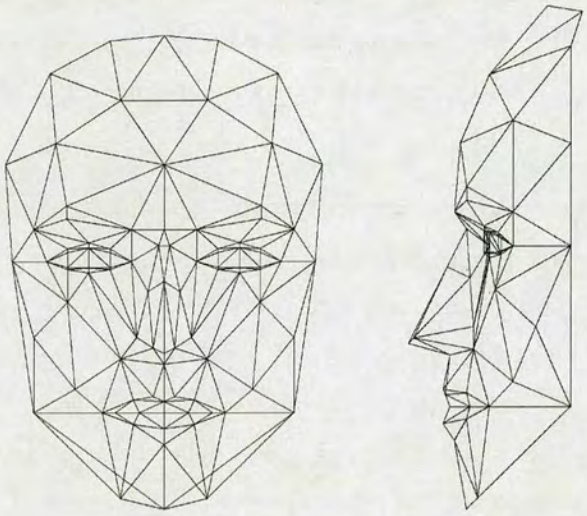


Figure 2.16: *The wire-frame model of Candide-3 [17]*

first and second sets, which are able to change the surface contour of the model, are called "Shape Units/Parameters" and "Animation Units/Parameters". The last set, only related to the model's rigid movement, is called "Global Pose Parameters". A "Shape Unit" (denoted by $\boldsymbol{S}$ in Equation 2.23) defines a facial trait such as head height and eye width, and its associated "Shape Parameter" (denoted by $\boldsymbol{\sigma}$ in Equation 2.23) is to augment or weaken such a facial trait.

Likewise, an "Animation Unit" (denote as $\boldsymbol{A}$ in Equation 2.23) describes a small facial movement such as jaw drop and outer brow raise, and its parameter (Animation Parameter is denoted as $\boldsymbol{\alpha}$) displays the "magnitude" of such a change in comparison with a neutral face [6]. Lists of "Shape Units" and "Animation Units", and their interpretation are given in Appendix C.1. Furthermore, in order to perform global motion, "Global Pose Parameters" are added for rotation ($\boldsymbol{R}$), scaling (s) and translation ($\boldsymbol{t}$). Therefore the Candide model could be described in the mathematical form below:

$$\boldsymbol{g} = \boldsymbol{R}s(\overline{\boldsymbol{g}} + \boldsymbol{S}\boldsymbol{\sigma} + \boldsymbol{A}\boldsymbol{\alpha}) + \boldsymbol{t} \tag{2.23}$$

where $\overline{\boldsymbol{g}}$ is the standard Candide model and $\boldsymbol{g}$ is the adapted/animated Candide.

[6] a neutral face is defined in [143]

Another advantage of using Candide-3 is that the model now is compatible with MPEG-4 animation rules [143]. The vertices of the Candide-3 can be found in FFPs[7] and the Animation Units include FAPs [8]. The conversion between Candid-3 and MPEG-4 is summarised in Appendix B.

While the Candide model is frequently used in education and research purposes because of its simplicity and availability, higher complexity models can be created by VRML (Virtual Reality Modelling Language) and its successor X3D [148]. VRML is a standard file format for representing 3D objects in computer graphics. It defines vertices and edges for a 3D polygon along with the surface colour, image-mapped textures, shininess, transparency and so on. Fitting the higher complexity model to a face requires more detailed information of the face surface and this requires use of some expensive hardware/software (for example, 3D scanning system from Cyberware^R [149]), rather than a number of face images. The texture rendering of the model is usually undertaken by using OpenGl (Open Graphics Library), a 3D computer graphical language [150].

[7]Facial Feature Points (FFPs) provide spatial references for MPEG-4 face model [143]. For example, FFPs define eye corners, nose tip, hair line and so on.

[8]Facial Animation Parameters (FAPs) are something analogous to Animation Units of Candide, but are defined based on the study of minimal perceptible actions and are closely related to Ekman's Facial Action Coding System (FACS) [3].

Chapter 3
Lip Fitting

3.1 Introduction

In order to produce smooth and realistic facial animations, a face model must be fitted onto the face with high accuracy. Among various facial features, the first and foremost feature to be fitted is the lip. Lips have been considered the most prominent facial features as they are very important for facial expressions [3].

Accurate lip fitting is required for many applications. These include:

1. Model-based coding requires it for low bandwidth transmission.

2. Psychologists need it to understand facial expressions and human intentions.

3. Film-makers need it to produce high quality speech animations.

4. Finally, it is an important factor in improving speech recognition, particularly in a noisy environment. For this, it is essential to know the characteristics of the shape of the mouth in both spatial and temporal domains, and correct lip extraction is required so that specific phonemes can be recognised (i.e., lip-reading) [30].

Deformable Templates and Active Contour Models (Snakes) are the two most commonly used techniques in lip contour fitting. The work of using these two techniques for lip fitting has been reviewed in Section 2.2.2.

The lip fitting techniques used in this work are based on Active Contour Models (Snakes). This is because the deformable template approaches can not fit lips well in fine scale. Since the templates are created by some pre-set shapes, the freedom of deformation of the shapes is limited. The deformable template works better for the features which only have limited deformation, such as eyes. However, for mouths which are very highly deformable features, the performance of the deformable template is degraded.

Active Contours Models (Snakes), which have higher deformability, can overcome this prob-

lem. Further analysis of the lip boundary reveals that a traditional intensity-type snake is unable to fit the lips precisely as the intensity edge of the lip boundary is not sharp. This leads to a development of the "colour snake" which is capable of extracting the lip by matching colour and desensitising the lighting effect. Some colour snake algorithms have been discussed in Section 2.1.3.

The following sections describe two lip fitting algorithms based on colour active contour models. Both apply the balloon pressures introduced in Section 2.1.3. The earlier one incorporates a fixed colour model whereas the latter one uses an adaptive colour model.

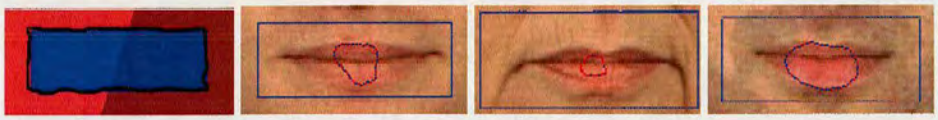
3.2 Initial Lip Fitting Model

This lip fitting algorithm was inspired by Schaub and Smith's work [36] (also discussed in Section 2.1.3). They based their colour snake on balloon snakes and used a Gaussian colour model in Hue-Saturation-Value (HSV) colour space. The pressure equations of their balloon used Hue and Saturation only in order to separate colour changes from illumination variations. They claimed striking results obtained in various lighting and shadowing conditions. However, the objects used in their experiments all contained obvious edges with high colour contrast across the boundaries, such as shown in Figure 3.1(a). An implementation of their technique failed to extract the lips as the lips possess the characteristics of slowly changing colour across the boundary. Figures 3.1(b), 3.1(c) and 3.1(d) show unsuccessful lip extraction by applying Schaub and Smith's approach. A lip fitting algorithm which retains the virtue of the use of colour but adds an additional intensity gradient term to enhance the sensitivity of the "edge" was therefore developed.

3.2.1 Lip Colour Model

A colour model containing sets of lip and non-lip examples is required. This enables the snake to understand the difference between desired and undesired colours and thus the object with desired colour can be extracted.

A suitable lip colour model should recognise all valid lip colours as desired colours while rejecting all unwanted non-lip colours such as facial hair or skin. The lip colour model was



(a) test example from [36] **(b)** subject 10311 **(c)** subject 10411 **(d)** subject 10511

Figure 3.1: *Figure 3.1(a), Schaub and Smith's approach is capable of extracting the inner boundary between the red and blue regions while the spurious edge resulting from the shadowing is ignored. Figure 3.1(b), 3.1(c) and 3.1(d), Show that their technique is unable to extract edges with slowly changing colour characteristics, such as lips. The dotted line is the final snake convergence.*

built based on Hue[1], using the fact that the skin and lip regions of individuals of different gender, age and ethnicity have major differences in intensity but not colour [151].

110 lip pixels from images in the XM2VTS database [16] were selected. These pixels were chosen from different lip regions of diverse facial images. As expected, the hue distribution of these pixels was approximately Gaussian with mean of 11.25 degrees and standard deviation of 4.01 degrees. A second set of 230 non-lip pixels from the same database was then selected. These pixels were also selected from a diverse range of images including those with facial hair. The best discrimination between these two sets (lips and non-lips) was found to be with a discrimination threshold of $\pm$ 1.5 standard deviations from the lips set. Using this threshold, 87% of the lip colours fell into this region and only 2% of the non-lips set were not rejected.

3.2.2 Active Contour Model

The original active contour models (snakes) and the balloon snakes have been reviewed in Section 2.1.3.

The active contour model selected is a modification of the pressure colour snake model proposed by Schaub and Smith [36]. An image gradient term was added, so that the snake can settle on an edge, even in an area with slowly changing colour characteristics.

[1]Saturation is not used as the experiment shows that saturation of the lip and skin is largely overlapped.

The energy function of this modified snake is thus given by:

$$E = \int_0^1 \frac{\alpha}{2}\left|v'(s)\right|^2 + \frac{\beta}{2}\left|v''(s)\right|^2 - \gamma\left|\nabla I\left(x(s), y(s)\right)\right|^2 - E_{\text{pressure}} ds \tag{3.1}$$

$x(s)$ and $y(s)$ represent x and y coordinates along the snake $v(s)$ and $I(.)$ is the intensity of the image. In Equation 3.1, the first two terms are the snake's internal energy, where α and β are weighting parameters which control its tension and rigidity. The last two terms represent the snake's external energy. The first of these is the image intensity gradient (this is the new term not appearing in Schaub and Smith's approach [36]) with a weight parameter γ applied. The last results from applying the pressure equation of the modified balloon model.

$$\nabla E_{\text{pressure}} = F_{\text{pressure}} = \rho\left(\frac{\partial v}{\partial s}\right)^{\perp}(\epsilon - 1) \tag{3.2}$$

where

$$\epsilon = \frac{|H(v) - \tau|}{k\sigma} \tag{3.3}$$

Equation 3.2 is the pressure applied perpendicularly to the derivative of the snake curve $v(s)$ with weight parameter ρ. Depending on the values of ϵ, the pressure can be inward causing the snake to shrink or outward causing it to expand. ϵ is calculated using Equation 3.3. Equation 3.3 is a simplified version of Equation 2.21. Since only the Hue (H) is considered, the square root in Equation 2.21 is replaced by an absolute operator to increase the computation speed. Again, τ, σ and k are mean, standard deviation (s.d.) and tolerance of the lip hue. $H(v)$ denotes the hue of the pixels on which the snake lies.

As can be seen in Equation 3.3, ϵ depends on the Hue similarity based on the Gaussian colour (hue) model described in Section 3.2.1. The numerical values found in Section 3.2.1 are used, $\tau = 11.25$, $\sigma = 4.01$ and $k = 1.5$. The larger the difference between $H(v)$ and τ is, the stronger pressure force is produced, and the faster the snake will shrink or expand.

Since the colour model is based on hue, it offers good discrimination between lip and non lip colours and the snake should eventually settle where there is a lip colour on one side and a non-

lip colour on the other. Moreover, in order to ensure an "edge" will be extracted, the weighted image intensity gradient term (the third term in Equation 3.1) has been incorporated to ensure some of the desirable characteristics of an edge. The lip is fitted by minimising the energy function (Equation 3.1). This can be solved by using the Euler-Lagrange equation [152] and a finite differences approach (see Section 2.1.3).

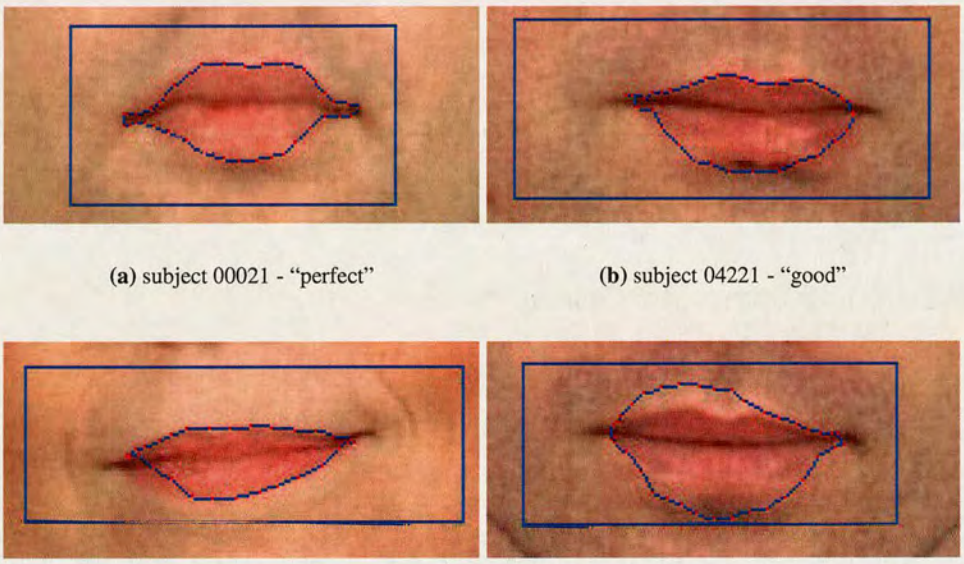
3.2.3 Results from Active Contour Model

This algorithm was tested on the XM2VTS database [16]. For this, a subjective grading system was created to assess the quality of lip fitting since a ground truth comparison was unavailable. The system graded the fitting into 5 different grades- "perfect", "good", "fair", "poor" and "wrong"- depending upon the general shape appearance and fitting of 4 benchmark points (left, right, top and bottom corners of lips). A score was also assigned to each grade; "perfect"=5, "good"=4, "fair"=3, "poor"=2 and "wrong"=1. Some examples of typical lip fitting grades are illustrated in Figure 3.2. As can be seen, subject 00021 is graded as "perfect" since the corners and edges of the lip are fitted precisely. By contrast, subjects 04221, 02021 and 00411 have lower grades since one or more corners or edges of the lips are not fitted well. Table 3.1 provides the percentage of lip fits in each category and the average scores for images with and without facial hair. Unsurprisingly, it is more difficult for the snake to extract the lips on images with facial hair. This is because facial hair often occludes the mouth region, making the lip edges invisible. Facial hair also contains a very wide range of non-skin colours which are difficult to be included in the colour model.

	Perfect (5)	Good (4)	Fair (3)	Poor (2)	Wrong (1)	Ave. score
Including facial hair	23%	36%	16%	11%	14%	3.43
Excluding facial hair	29.5%	46.2%	15.4%	5.1%	3.8%	3.92

Table 3.1: *Summary of lip fitting results*

As shown in Table 3.1 and Figure 3.2, even for images with no facial hair, there is still a significant proportion (24.3%) of fittings not achieving either good or perfect grades. More careful study of these images reveals that this is caused by the use of a fixed colour model. Lip and skin colours vary significantly from one person to another and can even vary for a given



(a) subject 00021 - "perfect" **(b)** subject 04221 - "good"

(c) subject 02021 - "fair" **(d)** subject 00411 - "poor"

Figure 3.2: *Examples of lip fit grading.*

individual. A fixed Gaussian hue model for all images is unable to cope with these variations. Since the hue distribution of each individual image is different, this must be taken into account in order to obtain better lip fitting. Therefore the system which can automatically adapt the hue model for each image was investigated.

3.3 Improved Lip Fitting Model - Active Contour Model with Adaptive Colour Model

The new lip fitting system was developed using an active contour with an adaptive colour model. The system is capable of automatically updating the parameters of the colour model, making the lip extraction more robust for various types of skin and lip tones. The overall system flowchart is depicted in Figure 3.3.

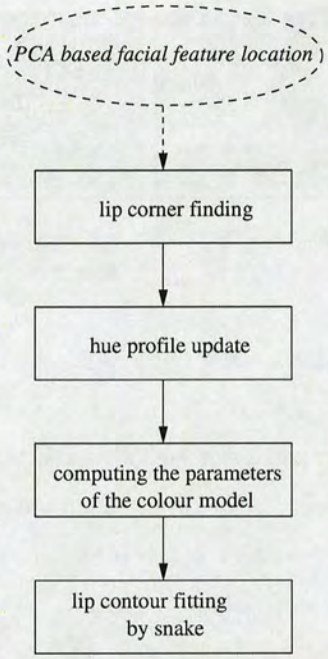


Figure 3.3: *Flowchart of lip fitting algorithm*

3.3.1 Lip Corner Finding

Initially, the system applies Hillman's technique [4] to estimate locations of important facial features. It starts with skin detection to an image to locate the face. Then Principal Component Analysis (PCA) with appropriate facial geometry techniques is applied to the face image to estimate the locations of the important facial features as an initial step in model fitting [4]. Details of this technique have been reviewed in Section 2.2.1. This enables location of suitable search regions for lip corners on the left and right hand sides of the PCA found mouth centres. Each of these search regions is 25×25 pixels, and is located in an area derived from PCA located eye and mouth centres.

It is important to find accurate lip corners as they will be used extensively in later stages. First, they are fiducial points for building the colour model. Second, the corner points can be made use of for snake initialisation. Finally, the found lip corners help the snake to extract the corner regions of the lip in order to achieve better lip fitting results. A lip corner finding algorithm similar to that proposed by Delmas et al. [93] is applied to the selected search regions (Figure 3.4(a)). A Sobel edge detector is applied to the hue images of these search regions to extract edge points of the lip around the corner (Figure 3.4(b)). A cost function is then employed to find

(a) lip corner search region **(b)** hue edge map **(c)** thinned hue edge map **(d)** found lip corner

Figure 3.4: *Procedure of lip corner finding*

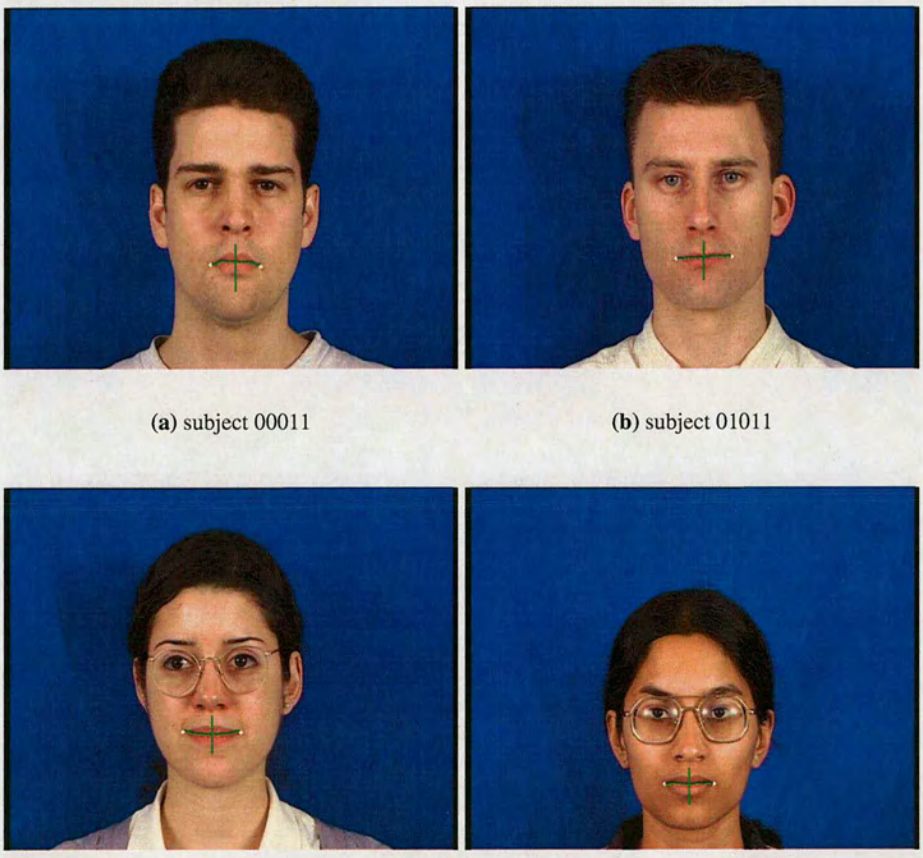
the boundary between the upper and lower lips (the grey curve in Figure 3.4(c)). An additional thinning operation is applied to Figure 3.4(b) so that more recognisable lip edges are revealed as shown in Figure 3.4(c). Finally, the lip corner is identified as the midpoint of the joins of the lip boundary and upper and lower lip edges.

After both lip corners are found, the horizontal and vertical mouth centre lines, are drawn for use in building the colour model. The horizontal mouth centre line connects the two lip corners along the darkest valley, while the vertical mouth centre line is a straight vertical line bisecting the lip corners. Figure 3.5 shows these two fiducial lines and lip corners in several sample face images.

3.3.2 Hue Profile Update

The vertical mouth centre line is used to construct the hue profile. This line is expected to run through the thickest region of the lip, which contains samples of the upper skin, upper lip, lower lip and lower skin. A hue profile of skin-lip tone is built by collecting pixels' hue values along this line. These hue profiles (as shown in Figure 3.6) typically show a U-shape. As the pixels are sampled from top to bottom of the lip, the left and right high levels in the profile represent the hue of the skin above the upper lip and below the lower lip, respectively, while the trough represents the hue of the lip region. This U-shape characteristic is as expected, since the skin hues are higher than the lip hues which are closer to red in the colour spectrum.

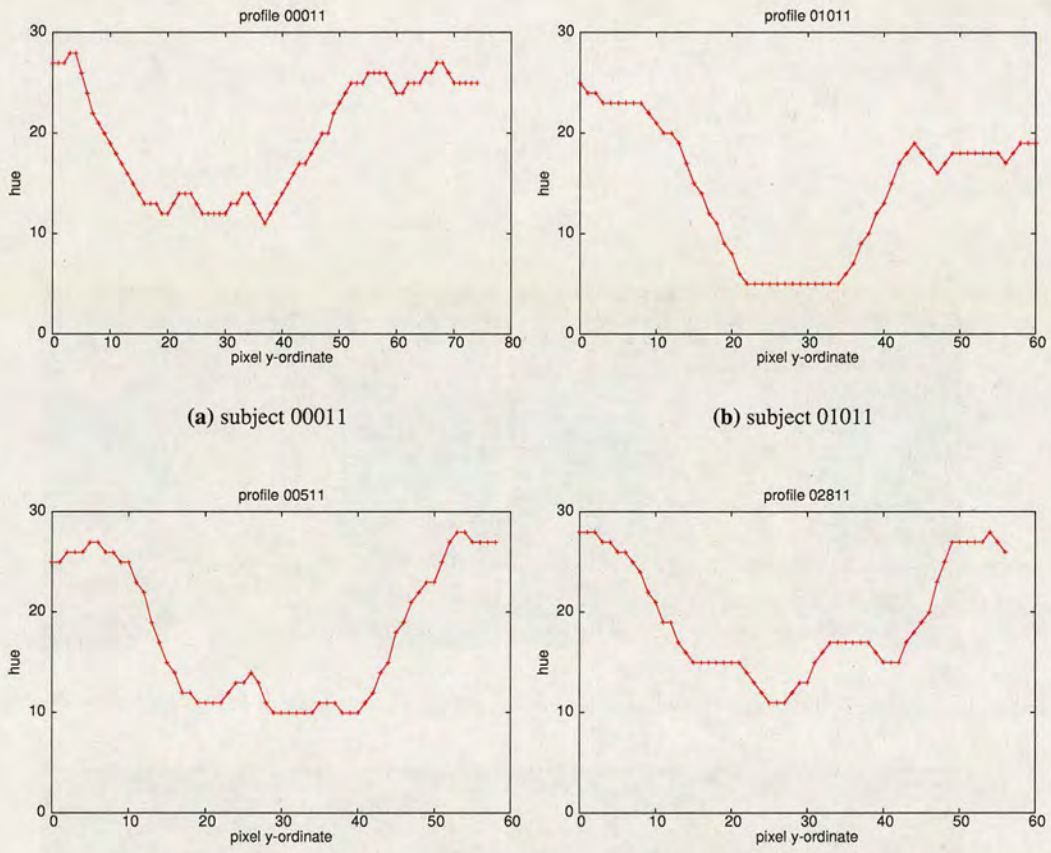
Although each profile shows a U-shape characteristic, a number of differences can be observed in Figures 3.6(a), 3.6(b), 3.6(c) and 3.6(d). For example, levels of the plateau and the trough of the U-shape, width of the U shape and steepness of the transition slopes are different among



(a) subject 00011 **(b)** subject 01011

(c) subject 00511 **(d)** subject 02811

Figure 3.5: *Sample images showing vertical and horizontal mouth centre lines and crosses marking the lip corners.*



(a) subject 00011

(b) subject 01011

(c) subject 00511

(d) subject 02811

Figure 3.6: *Hue profiles corresponding to the images in Figure 3.5*

the profiles. These differences confirm that a fixed Gaussian colour model used in the initial fitting algorithm (Section 3.2) is inadequate for all lip images.

3.3.3 Computing the Parameters of the Colour Model

The updated hue profile is used to adjust the parameters of the colour model so that the model can be adapted to individual images. A simple "dual threshold colour model" is chosen, which requires threshold values for the upper lip and lower lip boundaries and two weighting parameters associated with these thresholds. It is important to have separate thresholds for upper and lower lip boundaries, since the lower and upper skin hue can be quite different (for example, Figure 3.6(b)). In principle, the thresholds should be chosen to exactly separate skin and

lip hues but it is difficult to select such points from the hue profile since colour is gradually changing across the lip boundaries. In practice, however, the thresholds need not be absolutely accurate, since they are primarily used to attract the snake to approach the boundaries of the lip. The external force driven by the edge characteristics of the snake will then take over from this point and push the snake right onto the edge.

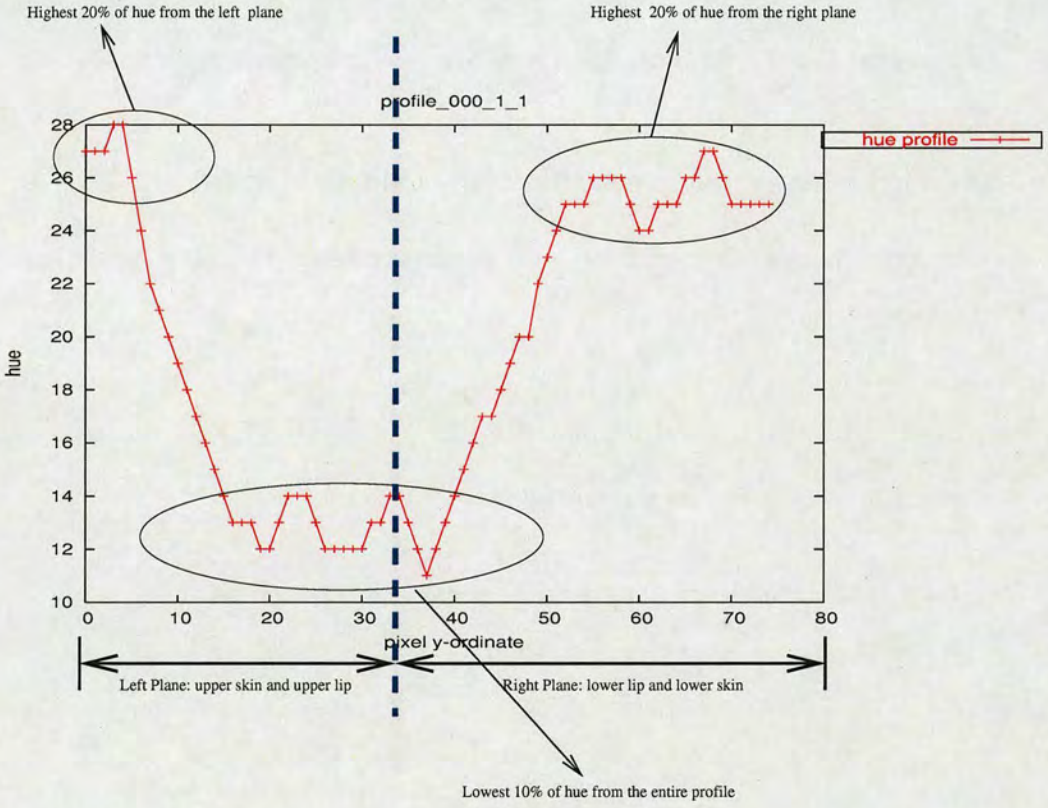


Figure 3.7: *The analysis of the hue profile of subject 00011*

An effective, adequately accurate, and automatic framework to compute the thresholds for upper and lower lip boundaries was developed. Firstly, the profile is split into two regions (Figure 3.7), the left and right plane, by the horizontal mouth centre line. Thus the left plane contains the hue of the upper skin and upper lip and the right plane contains the hue of the lower lip and lower skin. The upper and lower lip thresholds are calculated by applying the criteria below:

- the hue of the skin above the lip is computed by averaging the highest 20% of hue values from the left plane.

- the hue of the skin below the lip is computed by averaging the highest 20% of hue values from the right plane.
- the hue of the lip is computed by averaging the lowest 10% of hue values from the entire profile.
- the upper lip threshold ($Thr_{upper\ lip}$) is the average of the upper skin hue and lip hue.
- the lower lip threshold ($Thr_{lower\ lip}$) is the average of the lower skin hue and lip hue.

3.3.4 Lip Fitting by Colour Adaptive Active Contour Models

To incorporate the new colour model with the snake, the equation for calculating the pressure term F_{pressure} in Equation 3.2 is replaced by:

$$F_{\text{pressure}} = \rho \left(\frac{\partial v}{\partial s} \right)^{\perp} \cdot \epsilon \tag{3.4}$$

The error term ϵ in Equation 3.4 is computed for two scenarios:

if $v(s)$, the snake control point, is above the horizontal mouth centre line,

$$\epsilon = (H(x(s), y(s)) - Thr_{upper\ lip}) \times W_{upper\ lip} \tag{3.5}$$

else,

$$\epsilon = (H(x(s), y(s)) - Thr_{lower\ lip}) \times W_{lower\ lip} \tag{3.6}$$

$H(x(s), y(s))$ is the hue of the pixel on which the snake lies. $Thr_{upper\ lip}$, $Thr_{lower\ lip}$, $W_{upper\ lip}$ and $W_{lower\ lip}$ are the upper lip threshold, the lower lip threshold and the weights associated with the thresholds, respectively. The if-else clause in Equation 3.5 and 3.6 examines whether the snake control point $v(s)$ is above or below the lip; thus either the upper lip threshold or lower lip threshold is used accordingly. As shown in Equation 3.4, the inward or outward pressure, which causes the snake to shrink or expand is now directly proportional to ϵ. $W_{upper\ lip}$ and $W_{lower\ lip}$ are parameters to control how fast the snake shrinks/expands toward the upper or lower lips and are set to unity in the experiments for simplicity.

Using this adaptive colour model obviates the requirement to manually collect lip and non-lip

pixels from a large number of representative facial images. Also there is no need to compute mean (τ), standard deviation (σ) and estimate a suitable value for the discrimination threshold k. Since the discrimination threshold k usually needs to be carefully adjusted to obtain optimal lip fitting results, this approach represents a great improvement.

In conventional active contour methods, the initialisation of the snake can have a very significant effect on overall performance. However, the initialisation procedure presented here is simple. Because the active contour model incorporates a balloon model, the snake can be initialised reasonably far away from the lip, avoiding the difficulties of more commonly applied edge detection techniques.

The lip corners found in the earlier stage can facilitate snake initialisation. As shown in Figure 3.8, a rectangular box, on which the snake initially lies is chosen. This box has its left and right boundaries on the corners of the lips and its height equal to twice the vertical distance between the PCA found nose centre and mouth centre. The box has a width which is equal to the width of the mouth, permitting the snake to extract the lip with minimal iterations. Furthermore, the control points on the lip corners are anchored when the snake evolves. This overcomes the general limitations of the convergence of snakes to high curvature corners.

Experimental results have shown that at least 95% of lips in our large test data set could be fully enclosed by this type of snake initialisation, (failures were mainly due to poor initial PCA feature location). Figure 3.8 outlines the initial and final positions of the snake for facial images in Figure 3.5.

3.3.5 Results of New Active Contour Model

Three data sets were used to test the performance of the new lip fitting system,

- Set1: Normal XM2VTS database
- Set2: Half-lit XM2VTS database
- Set3: Susie and Foreman sequences

(a) subject 00011 **(b)** subject 01011

(c) subject 00511 **(d)** subject 02811

Figure 3.8: *The initial position of the snake and its final convergence to the lip*

Lip Fitting in Normal XM2VTS Images

This is the same database used to test the initial lip fitting algorithm in Section 3.2.3. Table 3.2 below shows the statistical summary of the lip fitting results using the adaptive colour threshold technique.

	Perfect (5)	Good (4)	Fair (3)	Poor (2)	Wrong (1)	Ave. score
Including facial hair	51%	23%	12%	8%	6%	4.05
Excluding facial hair	62.2%	28.0%	4.9%	3.7%	1.2%	4.46

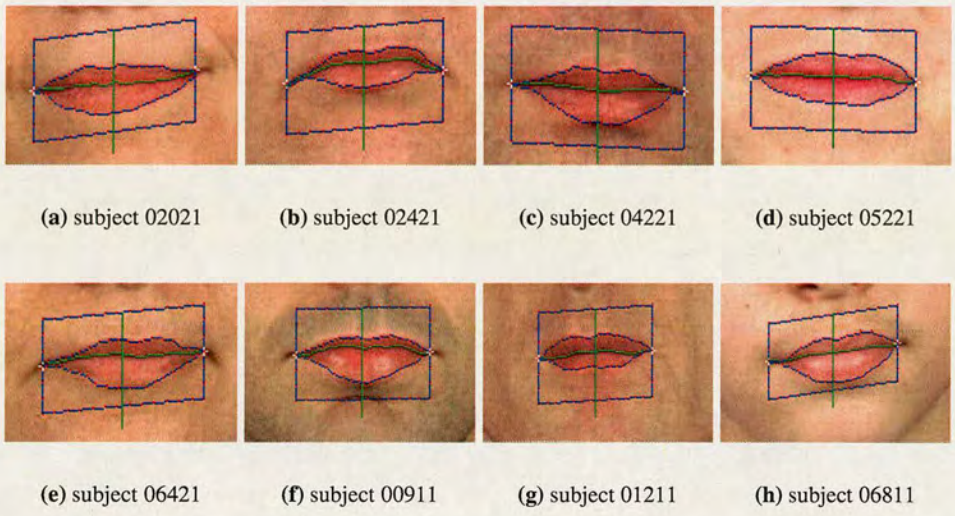
Table 3.2: *Summary of adaptive colour threshold lip fitting results.*

As can be seen, the percentage of "good" or "perfect" fitting has risen from 59% (see Table 3.1) to 74% for "including facial hair images" and from 76% to 90% for "excluding facial hair images", respectively. This improvement is due to the adaptive colour model being capable of selecting its parameters for each individual image so that better local lip fitting is achieved. Further analysis reveals that the percentage of "good" or "perfect" fitting rises to 98.6% if images with large areas of teeth revealed are excluded. This is because the hue profiles of these images are different so the lip threshold calculation is inaccurate. This issue will be addressed in Discussion section.

Figure 3.9 illustrates some close-up examples of the lip fitting for this database.

Lip Fitting in Half-Lit XM2VTS Images

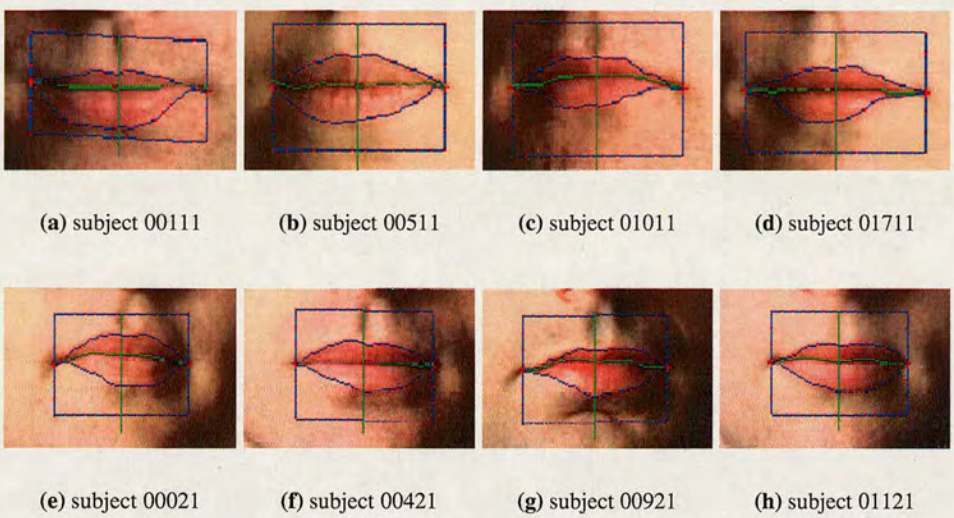
To further test the capabilities of the adaptive colour threshold lip fitting algorithm, a number of "half-lit" images from the XM2VTS database were used. These images were intentionally lit from only one side (either left or right) to produce unbalanced illumination. This provides a good simulation of a difficult situation likely to be encountered in real applications. Figure 3.10 shows that, even under these very uneven illumination conditions, the active contour can fit the lip fairly successfully.



(a) subject 02021 **(b)** subject 02421 **(c)** subject 04221 **(d)** subject 05221

(e) subject 06421 **(f)** subject 00911 **(g)** subject 01211 **(h)** subject 06811

Figure 3.9: *Lip fitting results of normal XM2VTS images.*



(a) subject 00111 **(b)** subject 00511 **(c)** subject 01011 **(d)** subject 01711

(e) subject 00021 **(f)** subject 00421 **(g)** subject 00921 **(h)** subject 01121

Figure 3.10: *Lip fitting results of half-lit XM2VTS images.*

Lip Fitting in Image Sequences

The algorithm was also applied to track the lips in the Susie and Foreman sequences [153]. A few modifications were made to enable the algorithm to track:-

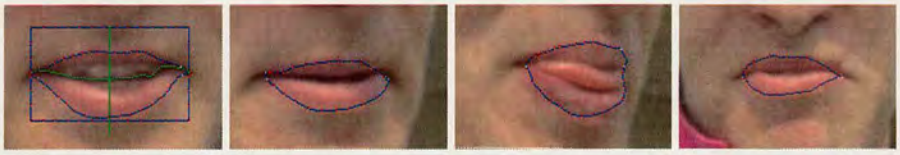
- (1) the lip corner finder is only applied to the first frame of the sequence to improve tracking speed.
- (2) by assuming no significant change in lip and skin tone of the same person in the sequence, the hue profile and colour model are based on the first frame with no further update for subsequent frames.
- (3) the snake control points near the lip corners are given smaller internal forces (i.e. by reducing the weighting factors α and β) to get better corner fitting.
- (4) the snake initialisation in each frame uses a 20% dilation of the snake convergence from the previous frame.

Figure 3.11 shows examples of the lip tracking for these sequences. Since the colour model is not updated during the sequence and the lip corner finder is only applied to the first frame (to speed-up the tracking), the results of the lip fitting are not as good as those for still images. However, the system is fully automatic and the tracking is not lost in the entire sequence.

This tracking algorithm requires roughly 1 second to process each frame in the sequence, thus it cannot be applied in real time. One way to increase the speed is to add a motion estimator, such as a Kalman filter. The issue of the fitting speed will be discussed in more detail in Section 7.2.

3.4 Discussion

As expected, facial hair and revealed teeth are two major factors which degrade the performance of the system. This section will assess how severely the system would be affected and possible solutions are suggested.



(a) foreman 000 **(b)** foreman 005 **(c)** foreman 010 **(d)** foreman 050

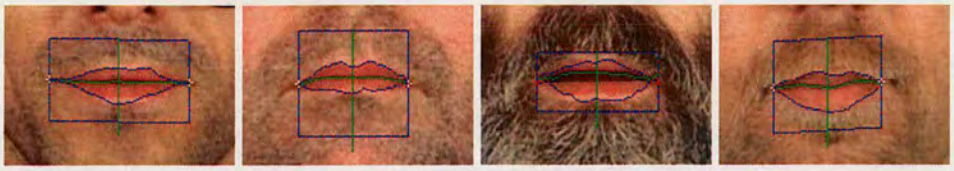
(e) susie 000 **(f)** susie 033 **(g)** susie 066 **(h)** susie 099

Figure 3.11: *Lip tracking in Foreman and Susie sequences. (a) and (e) are the first frames of the sequences. Lips in subsequent frames are tracked without colour model and corner updating.*

3.4.1 Facial Hair

Stubble does not cause problems to the system, as shown in Figures 3.12(a) and 3.12(b). Even in many cases of faces with beards or moustaches, the system can cope quite well (see Figures 3.12(c) and 3.12(d)). The system is most likely to fail in lip extraction when the beards/moustaches are very dark and dense, as shown in Figures 3.12(e) and 3.12(f). The reason is because the dark and dense facial hairs create strong intensity edges that may overpower the pressure force generated by the colour difference. This results in the snake settling on the edge between the facial hair and skin, rather than the lip edge. Figure 3.12(e) shows that the upper part of the snake is attracted to the edge between his moustache and the skin above it. Figure 3.12(f) shows the snake is attracted to the edge of his facial hair around the mouth rather than the true lip edge.

A possible approach to get around this problem is to identify the location of the facial hair and thus the colour properties of the facial hair can be known in advance of the lip fitting. The facial hair can be detected by using a texture recognition approach and thus coverage of the hair (whether or not it occludes the lip) can be estimated. Therefore the snake is required to fit either the lip-skin edge or lip-hair edge according to the hair occlusion to the lip.



(a) subject 02221 **(b)** subject 05111 **(c)** subject 00721 **(d)** subject 01621

(e) subject 01821 **(f)** subject 14111

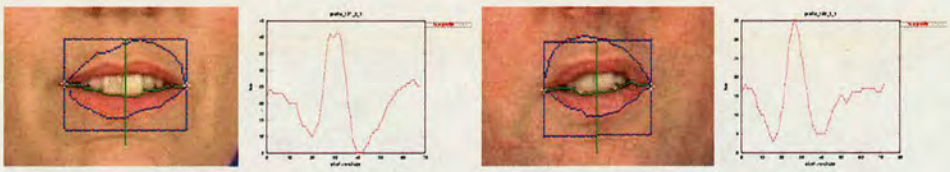
Figure 3.12: *Lip fitting on images with facial hair.*

3.4.2 Revealed Teeth

For a small area of visible teeth, such as those in Figures 3.13(a), 3.13(b), 3.13(c) and 3.13(d), the system is still capable of fitting the lips quite satisfactory. However, for large areas of teeth (Figures 3.13(e) and 3.13(g)), the hue profiles no longer show a U-shape characteristic (Figures 3.13(f) and 3.13(h)) and the criteria given in Section 3.3.3 are invalid. This leads to calculating wrong colour thresholds for the lip and skin, making the system unable to find the outline of the lip (Figures 3.13(e) and 3.13(g)).

To overcome this problem the system must identify the teeth and neglect the pixels from this area in the calculation of colour thresholds. On observation of the hue profiles of those images with large areas of revealed teeth, it seems that the teeth area corresponds to a very high hue peak in the profile (generally this peak is much higher than the skin hue). This may be because the teeth are whiter, meaning a small saturation value and thus a smaller radius in HSI colour space (recall Section 2.1.1). In consequence, a small colour variation would result in a large change in hue, thus possibly producing a large hue. Another more straight-forward approach may be to look at the Saturation profile. Therefore, a framework which makes use of the hue or

(a) subject 02211 **(b)** subject 0282 **(c)** subject 03321 **(d)** subject 04711



(e) subject 13721 **(f)** profile 13721 **(g)** subject 18011 **(h)** profile 18011

Figure 3.13: *Lip fitting on images with revealed teeth.*

saturation profile to identify the teeth area is suggested. Thus the teeth pixels can be removed from the threshold calculation and hence the correct thresholds should be obtained.

3.5 Conclusions

This chapter has described automatic lip fitting based on colour active contour models and introduced a new adaptive thresholding technique which gives improved performance over using a fixed Gaussian hue model. This enables the system to successfully operate with a wide range of subjects under different illumination conditions, offering improved performance over existing approaches. Successful lip tracking in well-known testing sequences has also been demonstrated.

Two factors which degrade the performance have been identified. Dense and dark facial hair produces strong edges between hair and skin and thus the snake is likely to be attracted by such edges. A texture recognition approach has been proposed to estimate the coverage of the hair so that the lip edge (either skin-lip edge or hair-lip edge) can be identified. Revealed teeth produces very different hue profiles so the threshold calculation may fail to obtain the correct colour thresholds. However this problem could be solved by identifying the teeth region from the profiles and ignoring teeth pixels in the calculation.

Chapter 4
Eye and Eyebrow Fitting

4.1 Introduction

For accurate model face fitting, features in the eye region must be located and properly fitted. This chapter concentrates on the fitting of the eye region - including fitting the eye itself and eyebrow. The term "eye fitting" is used to mean finding and locating the following eye components: iris, eye corners, and upper and lower eyelids, while "eyebrow fitting" is defined as drawing the outline of eyebrow contour.

4.2 Eye Fitting

4.2.1 Overview of the New Approach

Many researchers have applied Active Contour Model [6] and Deformable Template [7] approaches for eye fitting. Section 2.2.3 has given a review of these techniques. The eye fitting approach used in this work is based on Yuille's deformable templates [7] using parabolas and circles, but enhanced by exploiting the colour properties and eye corner information. Moreover, the updating procedure is simpler and more efficient: allowing one feature to be fitted properly before the next feature is fitted. This reduces the number of parameters to be updated simultaneously and offers more flexibility for template deformation. The iris is fitted initially (with a circle) and this is followed by fitting of the eye corners . These eye corner locations help with separate fitting of the upper eyelid and then the lower eyelid (with parabolic sections). Figure 4.1 depicts the overall system flow.

4.2.2 Iris Extraction

The eye fitting system is initialised by Hillman's approach [4] (also see Section 2.2.1). The skin detection technique and Principal Components Analysis (PCA) [4] are first applied to the image to locate the eyes. Then the further extraction and fitting of eye components can be followed.

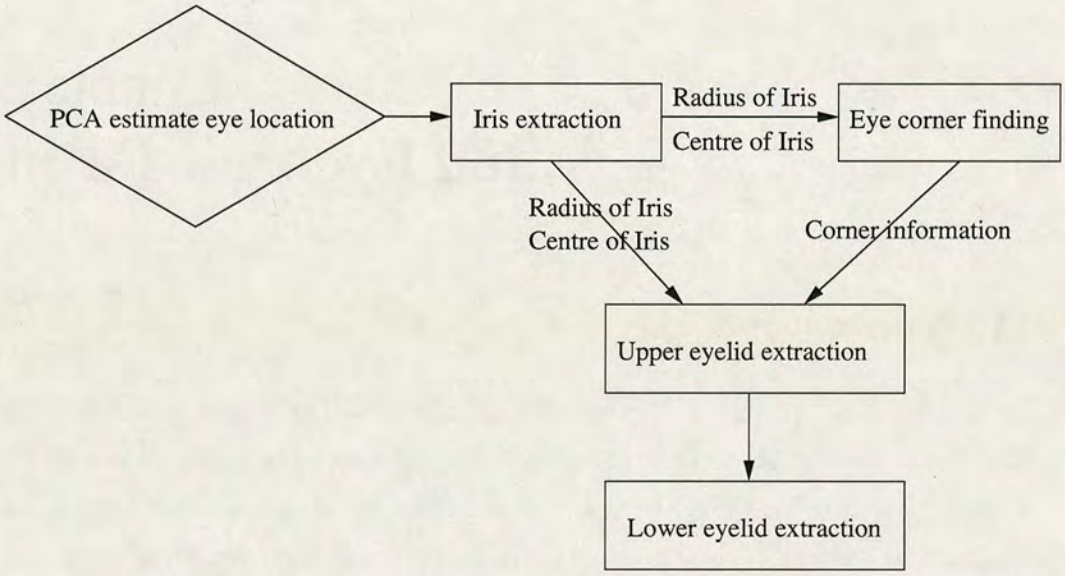


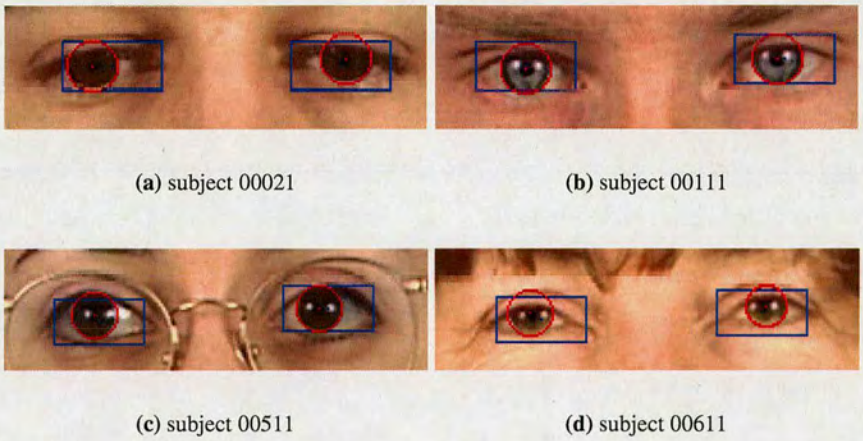
Figure 4.1: *Flowchart of the eye fitting system.*

The first (and easiest) eye component to extract is the iris. This is because the iris is round, dark, and has a large intensity contrast across its boundary. Starting with the PCA estimated eye locations gives a good search region for the iris.

Figure 4.2 shows a 40 $\times$ 20 pixel rectangular search region, centred at each PCA estimated eye centre. A circle of variable size (by changing its radius) is scanned across the search region (by changing its coordinate position) to find the best fit to the iris. The fitting process uses the Intensity Field, the Edge Field and the Radius of the Iris.

For colour images, the **Intensity Field** is simply $I(x,y) = \frac{1}{3} \times [R(x,y) + G(x,y) + B(x,y)]$ where $R(x,y)$, $G(x,y)$ and $B(x,y)$ is the intensity value in each R, G and B colour channel, respectively.

The **Edge Field** $\phi(x,y)$ for the iris is created in a slightly different way from Yuille's [7]. A Sobel edge operator is used to extract only the vertical parts of edges. This is because the upper and lower parts of the iris are frequently occluded by the eyelids so that only the sides of the iris may be visible (see Figure 4.2). In consequence, only the vertical parts of edges on the sides of the iris are taken as the iris boundary. The Sobel operator used for vertical edges is given as



(a) subject 00021 **(b)** subject 00111

(c) subject 00511 **(d)** subject 00611

Figure 4.2: *Examples of iris fitting.*

follows:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

To prevent the template from shrinking to the darkest spot inside the iris (the pupil), an additional size term is used. This is the expected **Radius of Iris** $R_{expected}$. The initialisation stage of PCA provides the approximate distance between the eyes and $R_{expected}$ is set to be one-tenth of this eye separation, on the basis of experimental observation.

Thus fitting the deformable circle to the iris requires maximising the following expression:

$$\frac{W_I}{A_{cir}} \sum_{(x,y)\in A_{cir}} [255 - I(x,y)] + \frac{W_E}{L_{cir}} \sum_{(x,y)\in L_{cir}} \phi(x,y) + W_S \times 255[1 - \frac{|R_{expected} - R_{deform}|}{R_{expected}}] \tag{4.1}$$

Where A_{cir} and L_{cir} are the area and circumference of the deformable circle. W_I, W_E and W_S are the weighting coefficients associated with intensity, edge and size terms, respectively. Since 24-bit colour images (8-bit in each channel) [16] are used in the implementation stage, "255" is the maximum value in Intensity and Edge Field. In consequence, the size term is multiplied

by 255 to give it the same scale as the other terms.

For a good fit, the area of the circle should be dark, a large intensity contrast should be present along the circumference and the size of the circle should be close to the expected value. Thus the first and second terms compute the average inverted intensity inside the circle and average edge intensity along the circumference. The last term provides a function yielding a high score as the radius of the deformable circle (R_{deform}) reaches $R_{expected}$. R_{deform} is allowed to vary from $0 \sim 2 \times R_{expected}$.

Figure 4.2 illustrates some iris fitting results. As can be seen, even for a relatively light coloured iris, this method can correctly fit the iris.

4.2.3 Eye Corner Detection Algorithm

Eye corners are very important features for eye fitting. If the eye corners can be found, this can improve the overall performance of the fitting and speed up the fitting of parabolas to the eyelids.

A simple, yet effective eye corner location technique is now described. This new eye corner finder, unlike that proposed by Lam and Yan [111], finds the inner and outer corners of each eye. It uses the white area of the sclera and corner templates.

A vector, with a length equal to three times the radius of the iris, and an origin at the centre of the iris is utilised. This length is chosen as it is long enough to cover the distance from the iris centre to the eye corner regardless of the position of the eyeball. This vector is fitted with an 'arrow head' which subtends a 30^o angle and has a length equal to the diameter of the iris. The vector, pivoted at the iris centre, rotates through $\pm 20^o$ with respect to the line between the two eye centres. The bottom of the arrow head is curved along the iris boundary. If the vector is pointing towards the eye corner, the area enclosed by the arrow head should contain mostly white sclera, i.e. the lowest saturation. Rotating the vector until the arrow head encloses the area of lowest saturation should point to the eye corner. Figure 4.3 illustrates such an eye corner arrow head arrangement.

Figures 4.4(a) and 4.4(b) illustrate the directions of the eye corners found using such a vector and arrow head approach.

Since the actual eye corner should be the point where the upper and lower eyelids meet, Corner

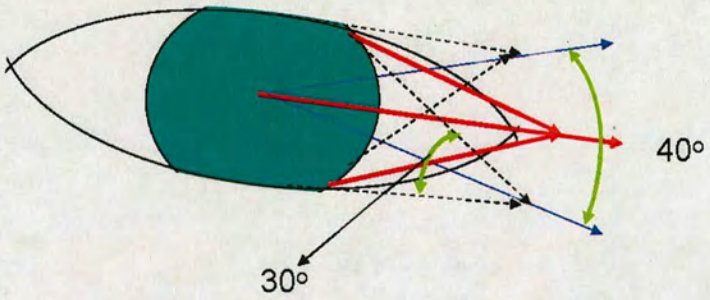


Figure 4.3: *Drawing of the eye corner arrow head*

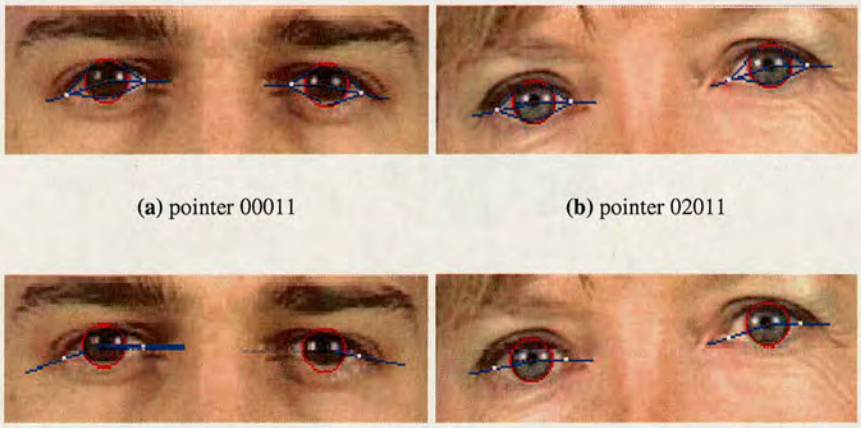
templates are incorporated in the calculation of the vector position to find such a point. These corner templates are given below; The two templates on the left are for the left eye corner (in fact, it is the object's right corner) while the templates on the right are for the right eye corner (object's left corner). The arrangement of numerical values in the templates finds the corner with a dark exterior and bright interior. Such a corner (e.g. eye corners) gives a large positive value when convolved with the templates.

$$\begin{bmatrix} -3 & -2 & -1 & 0 \\ -2 & -1 & 0 & 1 \\ -1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 \\ -3 & -2 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 3 & 2 & 1 & 0 \\ 2 & 1 & 0 & -1 \\ 1 & 0 & -1 & -2 \\ 0 & -1 & -2 & -3 \end{bmatrix}$$

These corner templates are incorporated in finding the direction and location of the eye corner. The eye corner is found when the following cost function is maximised.

$$\frac{W_S}{A_{arrow}} \sum_{(x,y)\in A_{arrow}} P[255 - S(x,y)] + W_C \times MAX\{E^+ + E^-\} \tag{4.2}$$



(a) pointer 00011 **(b)** pointer 02011

(c) corner 00011 **(d)** corner 02011

Figure 4.4: *Examples of finding eye corners. 4.4(a) and 4.4(b) show rotating vectors and arrow heads used to find the directions of the eye corners. 4.4(c) and 4.4(d) show the vectors and found corners (marked as white dots)*

$S(x,y)$	≥ 100	$75 \sim 100$	$50 \sim 75$	$40 \sim 50$	< 40
P	0.5	1	2	3	4

$S(x,y)$ denotes the saturation value at pixel coordinate (x,y). The first term in Equation 4.2 computes the average inverted saturation. Coefficient P is inserted to encourage lower saturation values while penalising higher saturation values.

The second term searches for the maximum value after performing a 2-dimensional convolution, along the vector, with the predefined templates shown in the previous page. E^+ is the convolution result using one set of the templates and E^- is its counterpart.

Finally W_S and W_C are the appropriate weighting coefficients associated with the saturation and corner template terms. Figures 4.4(c) and 4.4(d) show successful eye corner extraction examples.

4.2.4 Eyelid Extraction

Eyelids are the last features to be fitted since they take advantage of previously found features to achieve higher fitting accuracy. The upper eyelid is fitted first, followed by the lower eyelid.

Fitting Upper Eyelid

A parabolic section with parameters controlling its curvature, position of origin and rotation is employed to fit the upper eyelid. This deformable parabola is described in Equation 4.3. a controls the degree of curvature and can take values between 0 and $\frac{1}{\text{Radius of Iris}}$ (smaller a produces a flatter curve). (x_o, y_o) is the origin of the predefined axes of the parabola and x_m and y_m are subsequent horizontal and vertical translation of the parabola. Thus the origin of the shifted parabola becomes $(x_o + x_m, y_o + y_m)$. The parabola can also be rotated with respect to its origin. Equation 4.4 and 4.5 demonstrate how the parabola is rotated by an angle θ. D is the distance to the parabola origin from a parabola point (x, y) while α is the inclination of (x, y) with respect to the origin. $(x^{'}, y^{'})$ is then the new location transformed from (x, y) after the rotation operation. Figure 4.5 illustrates how the parabola is deformed and fitted to the upper eyelid.

$$(y - (y_o + y_m)) = a(x - (x_o + x_m))^2 \tag{4.3}$$

$$\begin{aligned} x^{'} &= D \times \cos(\alpha + \theta) \\ y^{'} &= D \times \sin(\alpha + \theta) \end{aligned} \tag{4.4}$$

$$\begin{aligned} D &= \sqrt{[x - (x_o + x_m)]^2 + [y - (y_o + y_m)]^2} \\ \alpha &= \tan^{-1}\left(\frac{\mathrm{y} - (\mathrm{y_o} + \mathrm{y_m})}{\mathrm{x} - (\mathrm{x_o} + \mathrm{x_m})}\right) \end{aligned} \tag{4.5}$$

The upper eyelid possesses the following characteristics; it presents a strong edge and an intensity valley, and the area below it, excluding the iris, is very white (sclera). In consequence, fitting the parabola to the upper eyelid requires maximising the following cost function:

$$\begin{aligned} &\frac{W_e}{L_{par}}\Big[\sum_{(x,y)\in L^o_{par}} \phi_o(x,y) + \sum_{(x,y)\in L^i_{par}} \phi_i(x,y)\Big] + \\ &\frac{W_v}{L_{par}} \sum_{(x,y)\in L_{par}} v(x,y) + \frac{W_w}{A_w} \sum_{(x,y)\in A_w} \omega(x,y) \end{aligned} \tag{4.6}$$

Where L_{par}, L^o_{par} and L^i_{par} are the lengths of the full parabolic section and the parabolic section outside and inside the iris, respectively. A_w is the white area defined below and W_e, W_v and

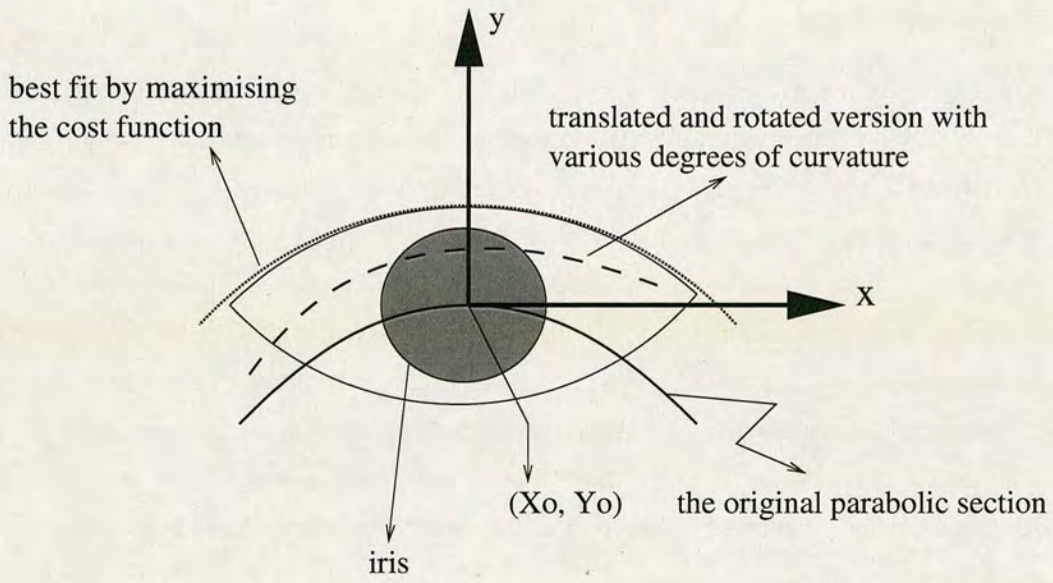


Figure 4.5: *Deformable parabola for eyelid fitting*

W_w are weighting coefficients associated with each term.

The first term in Function 4.6 represents an Edge term, containing the edge terms outside (ϕ_o) and inside (ϕ_i) the iris (if the iris is partially covered by the eyelid). ϕ_o is computed by a standard Sobel edge operator, while ϕ_i is computed by a horizontal Sobel edge operator, as an almost flat boundary is formed between the eyelid and iris when the eyelid cuts through the iris. The horizontal Sobel edge operator is given below.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The second term represents a Valley term which is introduced earlier. Since the upper eyelid stands out from the eye, it usually leaves a distinct shadow on the edge of the eyelid. Experiments confirmed that more robust fitting results are obtained if this term is included. $v(x, y) = 255 - I(x, y)$ and is simply the inverse of the Intensity. The final term represents a White Term (ω). This corresponds to drawing a horizontal line through the iris centre, intercepting the parabola at two points. The area (A_w) enclosed by the parabola and this line, but

excluding the part of the iris, should be white i.e., with low saturation. Thus ω is simply the inverse of the saturation, i.e., $\omega(x, y) = 255 - S(x, y)$.

To maximise Function 4.6 requires the parameters (curvature a, origin offsets (x_m, y_m) and rotation θ) of the parabola to be adjusted. Previously found eye corners are used to speed up the fitting process. Parabola candidates are only taken into consideration as they pass through the previously located eye corners or their nearest 4 neighbouring pixels. As a result, fitting is not a very time consuming process, despite the large number of parameters. Figure 4.6(a) shows an upper eyelid fit.

Fitting Lower Eyelid

The fitting of the lower eyelid is similar to that of the upper eyelid but uses a modified function.

The Valley term (v) has been removed from Function 4.6, as the shadow is absent from the edge of the lower eyelid. The White term has been modified so that the area where $\omega(x, y)$ is summed is bounded by the upper and lower parabolas but excluding the iris region. Finally, the curvature parameter a is set to $0 \sim -\frac{1}{\text{Radius of Iris}}$ in order to bend towards the correct direction.

Figure 4.6(b) shows the fitting on both upper and lower eyelids . The final stage of the fitting is to trim the parabolas and the iris so that only the sections between eye corners are left, as shown in Figure 4.6(c).

4.2.5 Experimental Results

The eye fitting algorithm was tested on the XM2VTS database. The weighting coefficients in Equations 4.1, 4.2 and 4.6 were set to unity during the test as it seems that every term is equally important. The origin of the parabola is initially situated at the found iris center, with curvature=0. For upper eyelid fitting, the offsets of the origin (x_m, y_m) are allowed to be varied within the range of $\pm 2 \times$ (Radius of Iris) and $0 \sim -1.5 \times$ (Radius of Iris) respectively. For lower eyelid fitting, y_m is varied within the range of $0 \sim +1.5 \times$ (Radius of Iris) so that the parabola can move down to fit the lower eyelid. The rotation of the parabola is within the range of $\pm 20^o$ to cope with some head tilt.

The outcome of the testing was encouraging. The technique was able to successfully extract

(a) (b)

(c)

Figure 4.6: *4.6(a) fitting an upper eyelid. 4.6(b) fitting both eyelids. 4.6(c) both eyelids fit after trimming.*

and fit a high percentage of eyes from this large data set (94% for iris extraction and 84% for eye corners and eyelids fitting). Figure 4.7 illustrates the fitting results on various types of facial images.

The successful fitting rate was not affected by factors such as age, gender or ethnicity, however, the system is vulnerable under the conditions of strong reflection and presence of spectacles. In the presence of spectacles, the successful fitting rate drops to 55%. This weakness will be addressed in the next section.

4.2.6 Discussion

As indicated previously, reflections and spectacles are two major causes of system failure. In fact, these two causes are related. Reflection occurring inside the iris causes wrong iris fitting as the iris selecting criteria of a "dark interior" becomes invalid. Reflection occurring elsewhere causes an unexpected region of lower saturation, which, again, interferes with the corner finding and eyelid fitting algorithm. The presence of spectacles worsens this problem, as spectacles usually increase the reflectivity and introduce extra frame edges.

Figure 4.8 shows eye fitting with various types of reflections and spectacles. Figure 4.8(a)

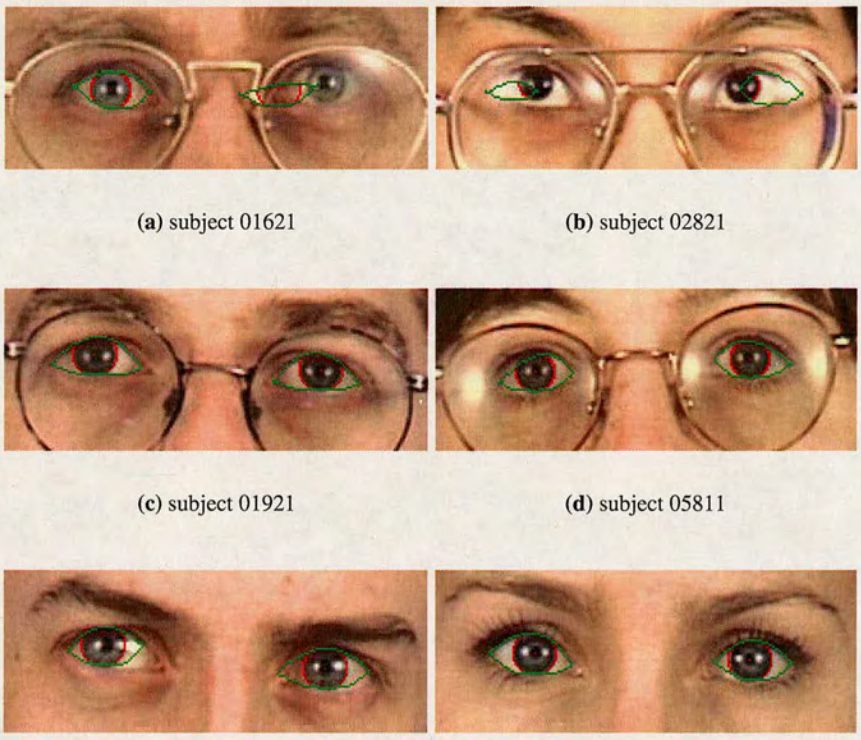
(a) subject 00121 **(b)** subject 00411

(c) subject 01311 **(d)** subject 02221

(e) subject 00321 **(f)** subject 00521

(g) subject 02011 **(h)** subject 02321

Figure 4.7: *Successful eye fitting on various types of faces*



(a) subject 01621 **(b)** subject 02821

(c) subject 01921 **(d)** subject 05811

(e) subject 07521 **(f)** subject 04111

Figure 4.8: *Eye fitting with reflections and glasses*

shows a wrong fitting to his left eye due to strong edges of the spectacles frame. Figure 4.8(b) has failed on both eyes due to very strong reflections on the glasses. Figures 4.8(c) and 4.8(d) have satisfactory fitting as no strong reflections are present in the eye region. Even though there are no glasses in Figure 4.8(e), the inner corner of the right eye is not found correctly due to a local reflection. Since there is less reflection inside the eye region in Figure 4.8(f), the eyes can be fitted well.

To reduce the interference from reflections and glasses, a reflection cancellation algorithm and a method to remove the spectacles frames is suggested. The work of Perez et al. [114] may be a good starting point for reflection cancellation. The spectacles frames can be detected using traditional edge detection approaches as the frame generally has very distinct edges.

4.3 Eyebrow Fitting

4.3.1 Overview of the New Approach

Despite eyebrows undergoing a great deal of movement when emotions are expressed, few researchers have considered the eyebrows in detail. Many researchers use low level techniques such as intensity thresholding and edge detection to segment eyebrows. These techniques are unlikely to be robust against different illuminations and a large variety of eyebrows. A review of eyebrow detection and extraction is given in Section 2.2.3.

The eyebrow fitting approach used in this work is inspired by Chen's work [127]. This approach uses an intensity-based k-means clustering followed by intensity snakes (Active Contour Models). Since all the operations work on the intensity of the image, lighting and shadowing effects need to be accommodated. Thus two additional steps are introduced which are capable of balancing the illumination of the image and removing the spurious shadows often formed near lower inner eyebrow corners. The algorithm begins by obtaining an image containing an eyebrow. The illumination is then balanced and the shadow is identified and removed from the k-means clustered mask image. Finally an active contour approach (snake) is applied to fit the outline of the eyebrow. Figure 4.9 depicts the system flow.

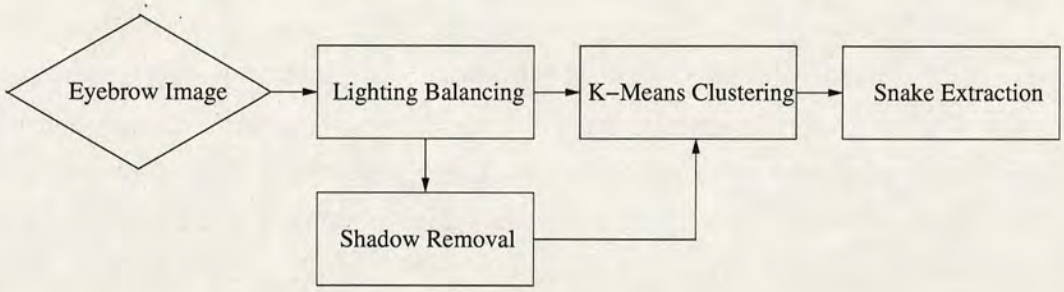


Figure 4.9: *Flowchart of the eyebrow fitting system.*

4.3.2 Lighting Balancing

Based on the eye contour fitting described earlier, a search region for an eyebrow can be located above the fitted eye with appropriate length and width. However, the illumination in the search region is often unbalanced transversely, due to the curvature of the forehead surface (see Figure 4.12(a)). To re-balance the lighting, skin and eyebrow tones are used as references. As illustrated in Figure 4.10, the eyebrow search region is divided into 5 equal vertical strips (i.e., Strip A, B, C, D, and E) and in each strip, a central vertical line is drawn (i.e., Linel, 2, 3, 4, and 5). The pixels' intensity is collected along those lines and so-called "eyebrow profiles" are plotted. Figure 4.11 shows such an eyebrow profile. Each curve corresponds to the intensity of a central vertical line. All curves, except for Linel and 5, exhibit a U-shape characteristic, indicating the vertical line has run from the skin above the eyebrow (high brightness), through the eyebrow (low brightness), to the skin below the eyebrow (high brightness).

To work out the average brightness of the skin and eyebrow on each central vertical line, it is assumed that the highest 50% of intensity of the line belongs to the skin and the eyebrow's intensity is estimated, according to its geometry, by the percentages shown in the bottom column of Table 4.1

pixels taken to average	L1	L2	L3	L4	L5
Skin: highest	50%	50%	50%	50%	50%
Eyebrow: lowest	10%	15%	20%	15%	10%

Table 4.1: *Proportions used to calculate mean skin and mean eyebrow intensity.*

Once the mean skin and mean eyebrow intensity are estimated on all the central vertical lines, the intermediate values between the lines can be obtained by linear interpolation. Assuming

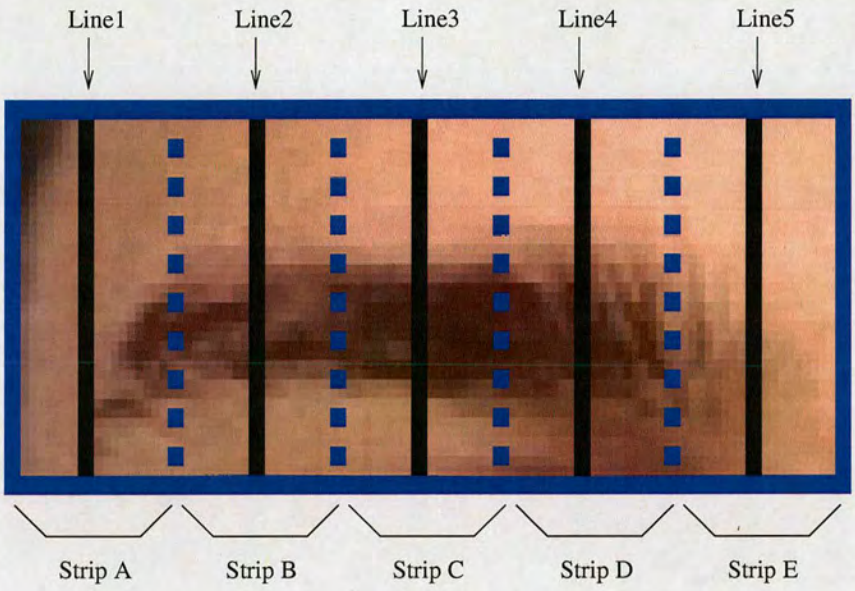


Figure 4.10: *The eyebrow search region is divided into 5 strips and 5 central vertical lines are drawn to obtain the eyebrow profile.*

the estimated mean skin and mean eyebrow intensity on each line is denoted by $\{I_{s1}, I_{eb1}\}$, $\{I_{s2}, I_{eb2}\}$, $\{I_{s3}, I_{eb3}\}$, $\{I_{s4}, I_{eb4}\}$ and $\{I_{s5}, I_{eb5}\}$, respectively. The mean skin and eyebrow in the area between Line N and Line $(N+1)$ is given by:

$$\begin{aligned} I_s(x) &= I_{sN} + \frac{I_{s(N+1)} - I_{sN}}{d} \times (x - x_N) \\ I_{eb}(x) &= I_{ebN} + \frac{I_{eb(N+1)} - I_{ebN}}{d} \times (x - x_N) \\ &\quad \text{when } N = 1, 2, 3, 4 \text{ and } x \in [x_N, x_{N+1}] \end{aligned} \tag{4.7}$$

where x_N is the x-coordinate of the Line N and d is the spacing between the central vertical lines i.e., $d = x_{N+1} - x_N$. The far left and far right regions in the search window are also estimated by linear extrapolation in a similar manner.

The images used are 8-bits per channel so the dynamic range of the intensity is 0 - 255. To balance the lighting, the mean skin intensity is arbitrarily shifted to 200 and mean eyebrow intensity to 100. This adjustment can be achieved by finding an appropriate scaling factor

$(SF(x))$ and offset constant $(OC(x))$ for every vertical line.

$$\begin{aligned} SF(x) &= \frac{N_{range}}{(I_s(x) - I_{eb}(x))} \\ OC(x) &= SF(x) \times I_s(x) - N_s \\ \text{thus,} \quad I_{adj}(x,y) &= SF(x) \times I(x,y) - OC(x) \end{aligned} \tag{4.8}$$

where N_{range} denotes the nominal range between skin and eyebrow intensities, which is $200 - 100 = 100$, N_s denotes the nominal skin intensity, which is 200. $I(x,y)$ is the intensity of the original image and $I_{adj}(x,y)$ is the intensity after the lighting balancing. Figure 4.12 shows an image before and after this lighting balancing operation. As shown in Figure 4.12(b), the skin region becomes more homogeneous after balancing.

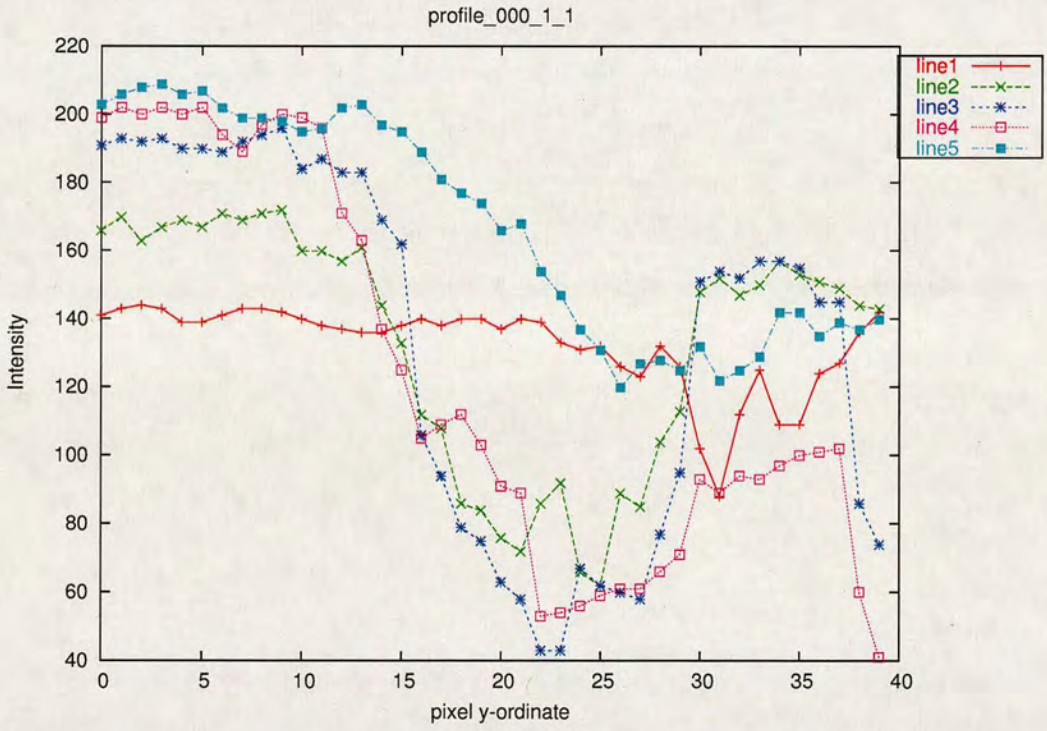


Figure 4.11: *The eyebrow profile for Figure 4.10.*

(a) before lighting balancing **(b)** after lighting balancing

Figure 4.12: *Image before and after lighting balancing. Notice that the intensity of the skin region becomes more homogeneous in Figure 4.12(b).*

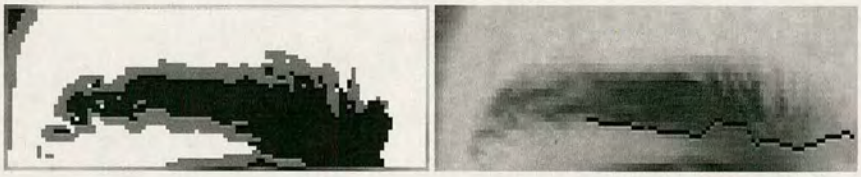
4.3.3 K-Means Clustering

K-means clustering is a non-hierarchical clustering technique so that the number of clusters, k, needs to be determined at the onset. Here, three clusters were used representing the skin, dense eyebrow and sparse eyebrow, respectively, as a two-cluster approach (one cluster for skin and the other for eyebrow) often misgroups the sparse part of an eyebrow as skin.

The initial cluster centroids were assigned according to two local traits - brightness and its variance. The initial centroid of the skin cluster was set as brightness = 200 and variance = 0 since the skin is of intensity of 200 nominally and homogeneous. The initial centroid of the dense eyebrow was set as brightness = 100 and variance = 0 since the eyebrow is of intensity of 100 nominally and homogeneous. Finally the initial centroid of the sparse eyebrow was set as brightness = 150 and variance = 100 since sparse eyebrow has a medium intensity and significant variance. A standard k-means routine is run to obtain the optimal classification for the desired clusters. A k-means clustered mask is generated and skin, dense eyebrow and sparse eyebrow is marked with intensity values of 255 (white), 127 (medium grey) and 0 (black). This 3-level intensity mask is shown in Figure 4.13(a).

4.3.4 Inner Corner Shadow Removal

A spurious shadow covering the inner corner of the eyebrow can be seen in Figure 4.13(a). This results from the fact that the eye is indented in the eye-socket and bones surrounding the eye protrude. To remove this shadow, a simple approach which assumes the shadow always occurs at the inner corner of the eyebrow is used. This proceeds by tracking a lower eyebrow boundary near the inner corner to separate the eyebrow and skin. This approach works for many common



(a) before removing shadow **(b)** lower boundary identified

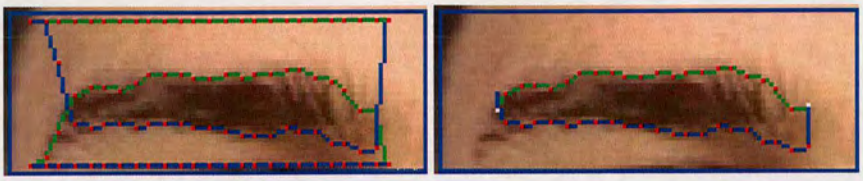
(c) after removing shadow

Figure 4.13: *The operation of shadow removal*

situations where the lighting is largely from above but would not be effective for other lighting arrangements.

The lower eyebrow boundary is very prominent at the middle part of the eyebrow and tails off towards the extremities. Thus the boundary is searched on the lower part of a vertical line bisecting the eyebrow window. The pixel which gives the maximum of Equation 4.9, indicating the presence of a strong intensity edge, can be found. This is the starting point of the lower eyebrow boundary and the next boundary point is searched horizontally towards the inner corner with a search space of +/- 2 pixels in the vertical direction. Figure 4.13(b) shows how such a lower eyebrow boundary is identified. Figure 4.13(c) shows that the inner corner shadow has now been removed. This is achieved by setting the intensity of the pixels below the found lower boundary to 255 (i.e., reclassified as skin).

$$MAX\{c1[I(x,y+2)-I(x,y+1)]+c2[I(x,y+1)-I(x,y)]+ \\ c2[I(x,y)-I(x,y-1)]+c1[I(x,y-1)-I(x,y-2)]\} \\ \text{where c1=1, c2=2} \qquad (4.9)$$



(a) before trimming **(b)** after trimming

Figure 4.14: *Eyebrow extraction by twin snakes*

4.3.5 Twin Snake Eyebrow Extraction

Two linear balloon snakes are now employed to extract the eyebrows on the k-means clustered mask after the removal of the inner corner shadow (Figure 4.13(c)). One snake is initialised near the top of the eyebrow search window, and progresses downwards to find the upper eyebrow boundary (green line in Figure 4.14(a)). The other is mirrored; initialised at the bottom and progresses upwards to find the lower eyebrow boundary (blue line in Figure 4.14(a)). Figure 4.14(a) illustrates initial and final convergence of the twin snakes. The eyebrow corners, which could be a problem for snakes due to the high curvatures, can be successfully extracted by identifying the intersection points of the two snakes.

The energy function of the twin snakes is defined by Equation 4.10. The first two terms are the internal energy, previously used in Equation 3.1. The last two terms represent the snake's external energy. The first of these is the potential associated with the balloon pressure whose magnitude and direction are determined by the average of the local intensities ($\bar{I}(x, y)$) of the image (Equations 4.11 and 4.12).

$$E = \int_0^1 \frac{\alpha}{2}|v'(s)|^2 + \frac{\beta}{2}|v''(s)|^2 + E_{\text{pressure}} + E_{\text{direction}} ds \tag{4.10}$$

$$\nabla E_{\text{pressure}} = F_{\text{pressure}} = \gamma(\bar{I}(x, y) - 127) \tag{4.11}$$

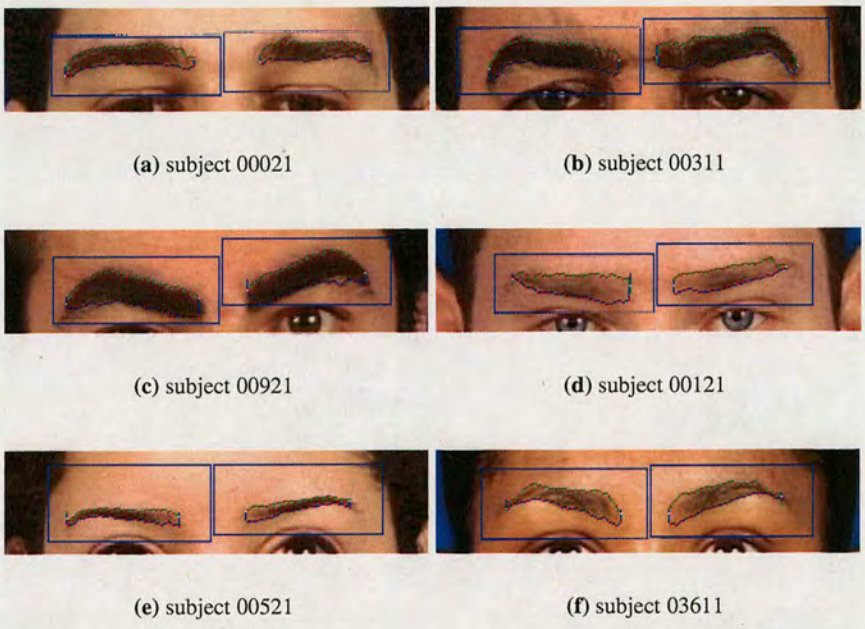
$$\bar{I}(x,y) = \frac{1}{9}\sum_{n=-1}^{1}\sum_{m=-1}^{1} I(x+n, y+m) \tag{4.12}$$

$$\nabla E_{\text{direction}} = F_{\text{direction}} = \delta(1 - \frac{2}{H}y)$$
$$\text{For upper snake, } y \in [0, \tfrac{H}{2}]. \text{ For lower snake, } y \in [\tfrac{H}{2}, H] \tag{4.13}$$

For the upper snake, since its initial position is in the upper skin area, (i.e., $\bar{I}(x,y)$ is above 127), this results in a pressure force pushing the snake downwards. Once the upper snake reaches the upper boundary of the eyebrow, the pressure force will reduce to zero (since $\bar{I}(x,y) \approx 127$). In the rare cases that the upper snake falls into the eyebrow region, the snake will be pushed back to the boundary due to an opposite pressure force (since $\bar{I}(x,y)$ now is below 127). The lower snake behaves in a similar manner. The last term of Equation 4.10 is the direction term which provides the guided directions for the two snakes to evolve. At the beginning, it produces a strong downward/upward force for the upper/lower snake to move to "clamp" the eyebrow. The force is gradually reduced as the snakes evolve and, if possible, is reduced to zero when the snakes reach the middle of the search window. This force is described in Equation 4.13. H is the height of the search window. The direction term increases the snake's robustness against noise, especially when there is hair present in the search window. γ and δ are the weighting parameters associated with pressure and direction forces. The outline of the eyebrow will be fitted by minimising energy function (Equation 4.10). This is solved by converting the energy function to a "force balancing function" through the Euler-Lagrange equation and then using a finite differences approach (see Section 2.1.3). The final extraction of the eyebrow can be obtained by trimming the redundant parts outside the found eyebrow ends (Figure 4.14(b)).

4.3.6 Experimental Results

The eyebrow fitting algorithm was tested on the XM2VTS database. It achieved a good fitting rate of 92% for the frontal-lit images without glasses and hair occlusion (see Figure 4.15). The algorithm was also tested on the data set of half-lit images to check if the lighting balancing algorithm could cope with such extreme conditions. It appears that the eyebrow contour in very dark regions cannot be fitted accurately (see Figure 4.16). Examples of eyebrow fitting results



(a) subject 00021 **(b)** subject 00311

(c) subject 00921 **(d)** subject 00121

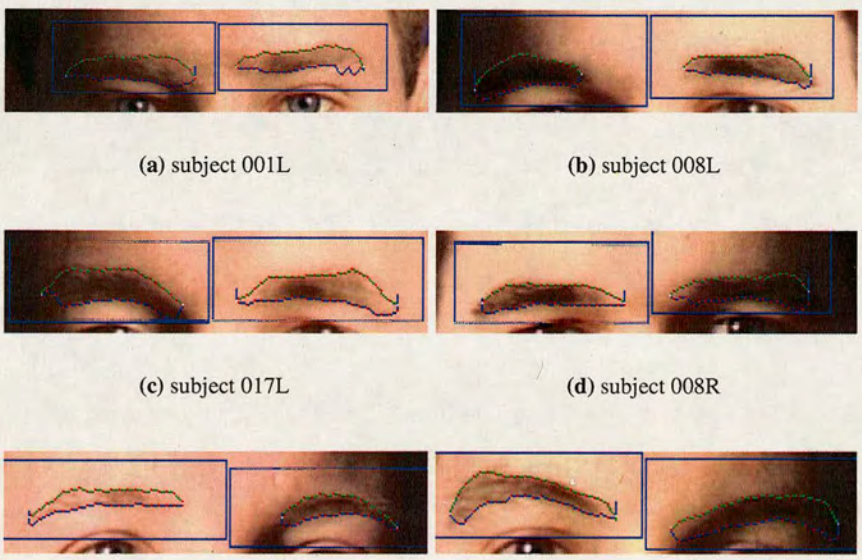
(e) subject 00521 **(f)** subject 03611

Figure 4.15: *Examples of eyebrow extraction in front-lit images*

are shown in Figures 4.15 and 4.16. The limitations of the approach have been identified as glasses, hair occlusion, and largely side-lit images. These limitations will be discussed in the next section.

4.3.7 Discussion

The presence of glasses or hair may occlude the eyebrow such that the snakes are unable to be fitted properly. Figures 4.17(a) and 4.17(b) show how the glasses interfere with the fitting. The lower boundary of eyebrow is not fitted successfully, instead, the edge of the glasses frame is fitted. The hair produces a similar problem (Figures 4.17(c) and 4.17(d)). As the snakes can not differentiate the edges of eyebrow and hair, the snakes are likely to attracted by hair edges before reaching the eyebrow. To tackle this problem, glasses frame removal and texture properties of the eyebrow could be used. The glasses frame can be removed by identifying the edge of the frame, which is normally much stronger than the eyebrow edge. The texture properties may be the major means to discriminate the eyebrow and hair, as the colour and intensity is obviously not suitable to use. The techniques [154–156] for converting intensity patterns into frequency components (Gabor filters and wavelets) seem to be promising to apply



(a) subject 001L **(b)** subject 008L

(c) subject 017L **(d)** subject 008R

(e) subject 011R **(f)** subject 021R

Figure 4.16: *Examples of eyebrow extraction in side-lit images. Figure 4.16(a), 4.16(b) and 4.16(c), light is from subjects' left hand side. Figure 4.16(d), 4.16(e) and 4.16(f), light is from subjects' right.*

for eyebrow-hair segmentation.

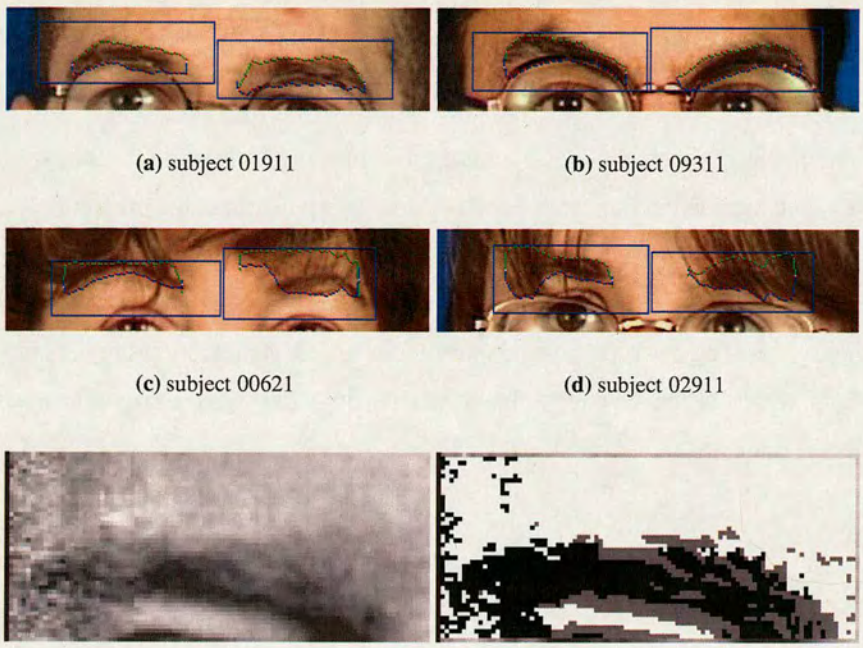
The performance of the eyebrow fitting is degraded in the side-lit images, indicating the linear lighting balancing scheme (introduced in Section 4.3.2) is unable to cope with such largely unbalanced lighting conditions. The reason is that since the intensity of the dark region of the side-lit image is very small, the scaling factor ($SF(x)$) computed by Equation 4.8 becomes very large. This means that not only the intensity of the eyebrow and skin is scaled up, but also the noise. This side-effect of noise augment is shown clearly at the right end (reader's left end) of the subject 001L's eyebrow (see Figure 4.17(e)). In consequence, the k-means clustering tends to be noisy (see Figure 4.17(f)) and thus the fitting of the eyebrow is degraded.

A possible solution for this is to design a more sophisticated lighting balancing scheme. The lighting balancing used in this work applies linear interpolation and shift can only compensate slightly unbalanced illumination. For these deliberately side-lit images, an analysis of light source and where the shadow will form may be required. In consequence, a non-linear lighting balancing scheme can perhaps be generated. Furthermore, the texture properties of the eyebrow can be helpful. As the shadow only affects the intensity value, the texture of the eyebrow should be preserved and can be identified in the dark regions.

4.4 Conclusions

This chapter has described improved approaches for eye and eyebrow fitting. The technique for eye fitting is based on Yuille's deformable template approach [7]. A circular template with a special size term is used for fitting the iris while parabolic templates are used for (upper and lower) eyelid fitting. A novel eye corner fitting technique employs a rotating "arrow head" which is pivoted at the found iris centre. The eye corner is found when the area of the arrow head contains most of white part of eye (sclera) and a strong corner characteristic (a jointed edge) is detected. The found eye corners not only increase the accuracy of the eyelid fitting but also speed up the fitting process as the the candidates of the parabolas must pass the found corners (or the neighbouring pixels). The fitting is in a cascade manner, that is, allowing one eye feature to be fitted properly before the next feature is fitted. This reduces the number of parameters to be updated simultaneously and offers more flexibility for templates deformation.

The limitations of this eye fitting approach include reflection and occlusion of spectacles. The reflection may be identified by using Perez et al's [114] technique to detect abnormal high (or



(a) subject 01911

(b) subject 09311

(c) subject 00621

(d) subject 02911

(e) lighting balancing of the right eyebrow of Figure 4.16(a)

(f) K-means clustering of the right eyebrow of Figure 4.16(a)

Figure 4.17: *Figure 4.17(a) and 4.17(b), eyebrow extraction with occlusion of the glasses. Figure 4.17(c) and 4.17(d), eyebrow extraction with occlusion of hair. Figure 4.17(e) and 4.17(f), the lighting balancing and k-means clustering applies on the right eyebrow of Figure 4.16(a). Both images are shown to be noisy.*

change of) saturation. The spectacles frame can be removed by using traditional edge detection techniques as the frame produces distinct edges. Eyebrows are fitted using a series of different techniques. A linear lighting balancing scheme is developed to compensate the unbalanced lighting due to the curvature of the forehead. This leads to a better k-means clustering which classifies the eyebrow search area into three regions -skin, dense eyebrow and sparse eyebrow. The shadow near the lower inner eyebrow corner is then removed by a lower inner eyebrow boundary detection. Finally two snakes are applied to fit the eyebrow contour.

The limitations of this eyebrow fitting approach include hair and spectacles occlusion, and heavy shadow. The spectacles may be removed by detecting the strong edges, while the hair may be removed by using the texture properties. Fitting the eyebrow in the heavy shadow resulting from largely side-lit images is challenging. A possible solution may be to use a more sophisticated (non-linear) lighting balancing scheme with exploitation of the texture properties.

Chapter 5
Chin Fitting

5.1 Introduction

To make a precise fit of the wireframe face to a real face, the face boundary needs to be located and boundary points need to be extracted. To achieve this, robust, accurate and automatic chin fitting techniques are developed. The techniques are based on Active Contour Models (Snakes) as they have been proved to be an effective approach for chin fitting (see Chapter 2). In this chapter, two chin fitting approaches are proposed. The initial approach, which uses an intensity snake, is capable of fitting the chin in a frontal-view, balanced-lit face image. The second approach, which continues from the initial one and exploits the symmetrical properties of human faces, is able to handle the chin in heavy shadowing and bad lighting conditions. Furthermore, an adaptive method for self-tuning the snake's parameters has been adopted in the proposed approaches. This self-tuning technique is simple and very effective in response to topological variations in the image. It is suggested that this self-tuning technique could be applied in other active contour model applications.

5.2 Chin Characteristics Analysis by Circular Profiling

Figure 5.1 shows a typical frontal face image. The chin contour can be divided into three different regions. In regions A1 and A2, the face is against the background. These regions are located on the both sides of the face. In region B, the face is against the neck skin. This occurs at the bottom part of the face. In regions A1 and A2, the chin edge is the face boundary and this part of the chin is easier to identify due to the substantial difference between the skin and background. On the other hand, the chin edge in region B is less definite because the face and neck possess very similar skin colour.

Further analysis of the chin characteristics can be achieved by drawing lines from the mouth centre with different angles. Figure 5.2 illustrates six lines drawn from the mouth centre at 36 degrees apart. The mouth centre is set to the midpoint between the two mouth corners found

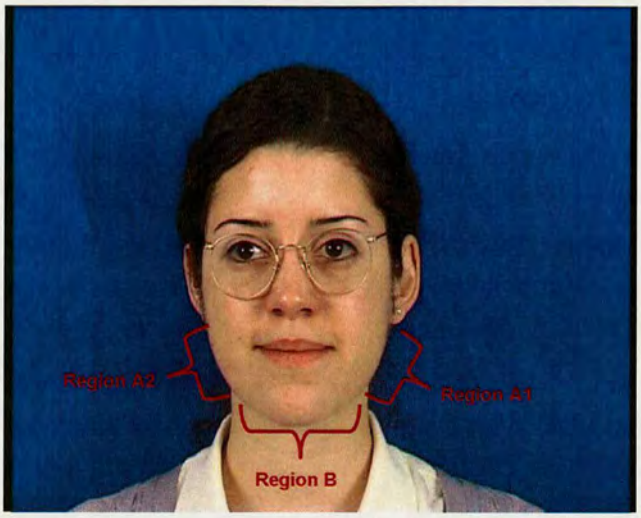


Figure 5.1: *An example of a frontal image showing different characteristics of the chin edge in different regions*

in the previous chapter. Thus the intensity of the pixels along each line can be collected and plotted against pixel distance from the mouth centre. This is called the "circular profile" of the chin contour.

Figure 5.3 depicts the circular profile of Figure 5.2. As can be seen, Line 1 (red) and Line 6 (yellow) fall abruptly near the end, indicating a face boundary (which is a chin edge) is reached. Line 2 (green) and Line 5 (light-blue) also show a sudden fall near the end; but in line 2 a small intensity valley is observed just before the final fall. By examining Figure 5.2, this intensity valley is likely to be the location of the chin contour. Line 5 does not have this small valley because the line is through the point where the neck and chin boundary meet (see Figure 5.2). As expected, Line 3 (dark-blue) and Line 4 (purple) do not display the sudden fall near the end. This is consistent with Figure 5.2 as the lines remain in the skin region (the neck). Instead, an intensity valley now shows a sign of the chin contour existing between the face and neck. Notice that the beginnings of the lines in the profile are quite ragged since they are affected by the lip region. These would not be used for chin extraction.

By studying the circular profiles, a chin characteristic can be identified. It is summarised as follows: If the chin edge occurs in regions A1 and A2 (Figure 5.1), an abrupt "fall" (or not a "fall" if the background is not dark) or something quite dramatic will be expected in the profile due to changing from one medium to another. If the chin edge occurs in region B (Figure 5.1),

an (gradual) intensity valley will be expected. Also, the region inside the lip should not be considered for chin search.

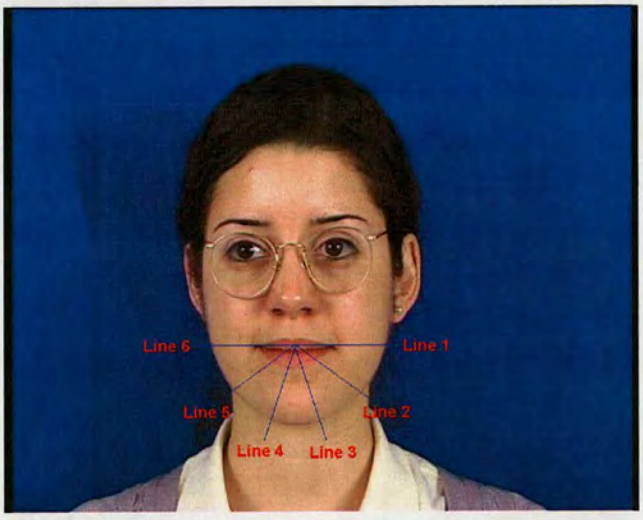


Figure 5.2: *Six lines drawn from the centre of the mouth. The circular profile of the chin is plotted by picking up the pixels along the lines.*

5.3 Initial Chin Fitting Approach

The flow-chart of the initial approach is depicted in Figure 5.4. The system starts with skin detection and utilises the lip corner points, which are found in the former chapter, to set up a search region for the chin. An Active Contour Model (Snake) can then be initialised on the boundary of the search region. The snake is an adaptive, inflating balloon, intensity snake, starting on the inner boundary of the search region and expanding to extract the chin contour by exploiting the chin characteristics found in the Section 5.2. This snake is able to self-adjust its own parameters, depending on the topology of the image, to obtain more robust chin extraction results. Each function block in the Figure 5.4 will be explained in more detail in the next sections.

5.3.1 Skin Detector

There have been many algorithms for detecting the skin in a face image in the literature [4, 49, 59–64]. The basic theory relies on the fact that skin from different people has similar chromaticity, although skin intensity can be quite different. Thus, it is possible to segment

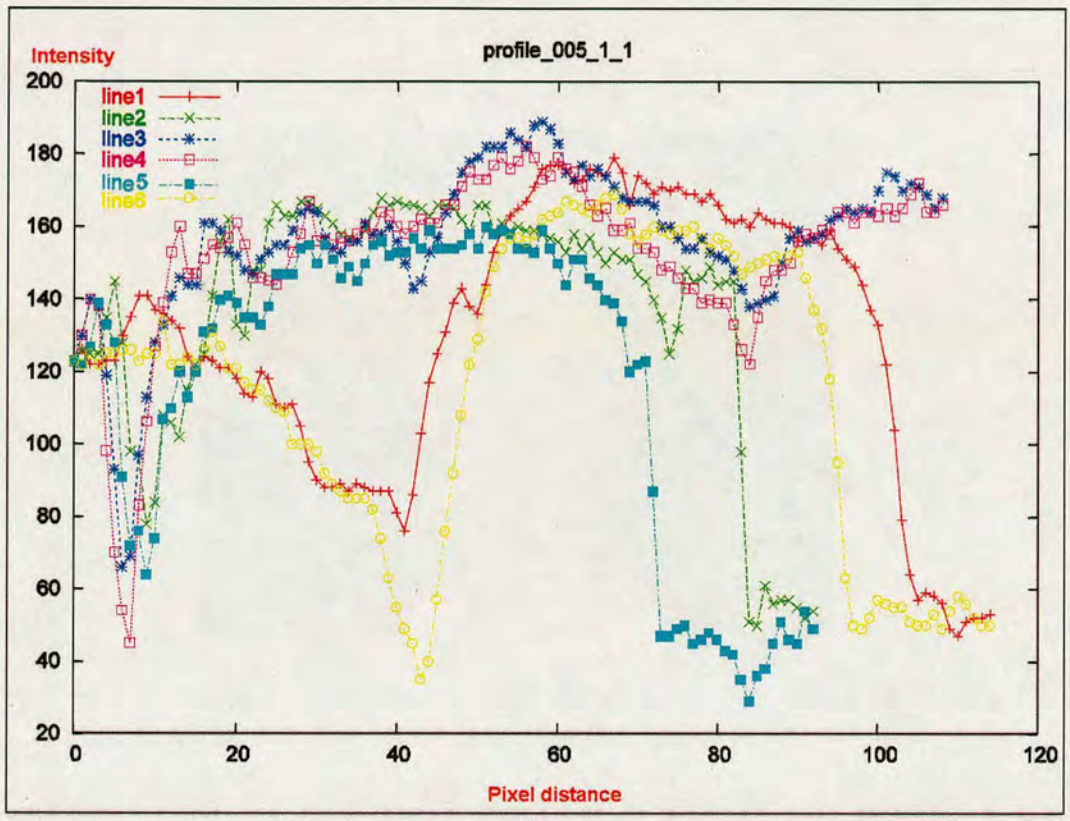


Figure 5.3: *The intensity profile corresponding to the lines in Figure 5.2*

the skin from the image by transforming RGB (Red-Green-Blue) values to a relevant colour space. A review for skin detection, thus face detection, is given in Section 2.2.1 A skin detector used here is to transform the image to HSV (Hue-Saturation-Value) space and impose a number of experimental thresholds on the Hue and Saturation channels, so that the skin region, which is meant to be different from other surfaces in Hue and Saturation, can be segmented. It then undergoes smoothing and another threshold procedure to obtain a black-and-white face mask image (Figure 5.5).

In this mask image, the "jaw points" can be located by extending a horizontal line from the mouth centre, thus the jaw points are found as the first black pixel on both sides. The mouth centre is set to be the midpoint between the two found lip corners. Of course, if the face is severely tilted, a horizontal line to search the jaw points may not be a perfect approach. However this can be easily cured by estimating the degree of the head tilt by considering the lip corner and eye positions. For simplicity, it is assumed that the head in the image does not

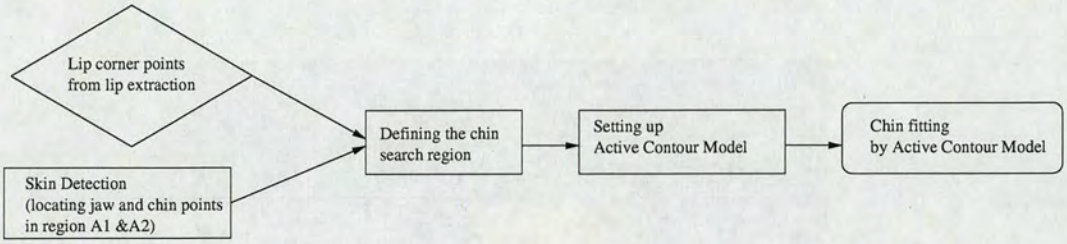


Figure 5.4: *The flow chart of the initial chin fitting approach*

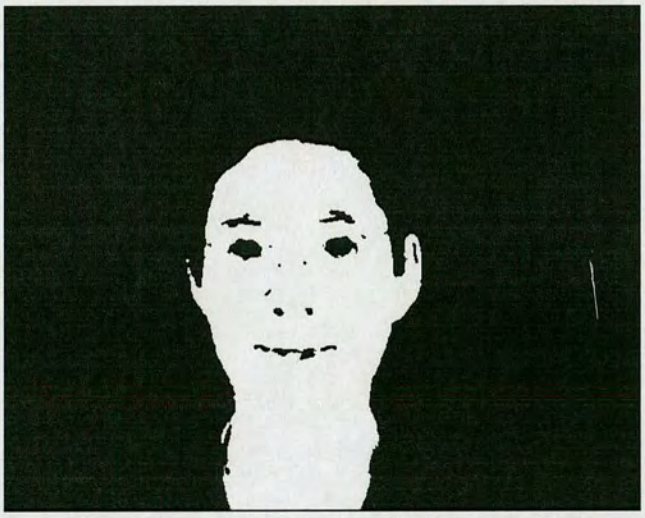


Figure 5.5: *The binary face mask is used to find the jaw points and the chin not backgrounded by the neck*

contain much rotation. Apart from the jaw points, the chin contour not backgrounded by the neck (refer to the region A1 and A2 in Figure 5.1) is also identified in this face mask image.

5.3.2 Defining Chin Search Region

A chin search region is a confined area between two concentric 180-degree arcs (centred at the mouth centre), and the face boundary found in the skin detection stage. This region is shown as a red-shaded area in Figure 5.6. The diameter of the outer arc (green dots) is set to be 1.2 times the jaw width, while the diameter of the inner arc (yellows dots) is set to be 1.2 times the lip width. The jaw width and lip width are the distances between the two found jaw points and two found lip corners, respectively. This experimentally set search region encompasses the

chins for 95% of the frontal images in the test database, XM2VTSDB [16].

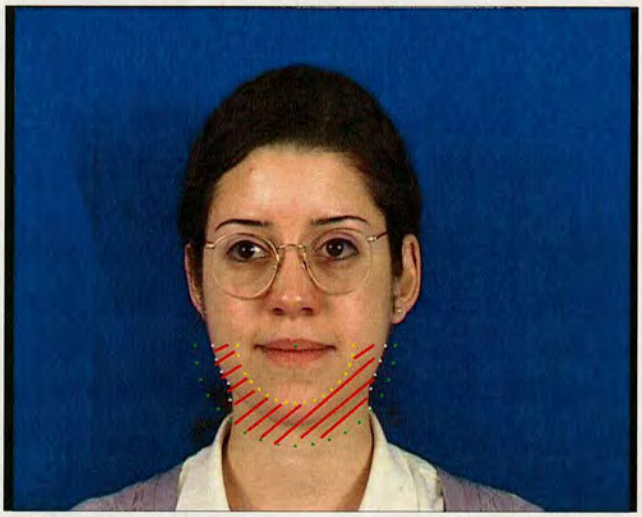


Figure 5.6: *The red-shaded area confined by inner and outer semi-circles and the face boundary is the chin search region*

5.3.3 Setting up the Active Contour Model

The proposed active contour model (snake) is initialised on the inner arc (yellow dots in Figure 5.6) and expanded outwards to find the chin contour. The snake can only move radially in a 1-D polar coordinate system with origin at the mouth centre. The snake is allowed to go outwards as far as reaching the outer arc (greens dots) or the face boundary (white dots). The reason for using an inflating snake rather than a deflating snake (which is adopted for the lip fitting) is that there are more undesired features outside the chin contour than inside. The background, hair, cloth and neck edges are very diverse and can interfere greatly with the snake's use for chin fitting. Starting from the inner arc guarantees that the growth of the snake is within the homogeneous skin region. It results that the chin is the only feature that needs to be modelled in the snake equations.

The snake contains 19 control points (shown as the 19 yellow dots in Figure 5.6 as they are deployed at the initial position), at 10 degrees apart. The number of the control points affects the final fitting results. If an excessive number of points is used, the chin fitting takes longer time and the result is sensitive to the noise. On the other hand, using an insufficient number of points can not provide enough detail of the chin. The best number was decided empirically and

is varied with the size of the face in the image.

This proposed snake works on a polar plane instead of the normal Cartesian plane. The plane origin is located at the mouth centre and the plane covers the chin search region. Two major benefits are obtained by using such a polar plane framework. First, all the control points are only allowed to move radially in the polar plane. In other words, the control points can only move directly towards (inwards) or directly away from (outwards) the mouth centre. This 1D (one-dimensional) movement of the control points reduces computational time greatly compared to a 2D snake. Second, a semicircle template is implicitly used as a shape constraint on the snake convergence. In the Cartesian plane, the minimisation of the snake's internal energy leads to a straight line or a point. In the polar plane, however, the internal energy is minimised when all the control points are situated at an equal distance to the mouth centre. It intrinsically imposes a constraint of a semicircular shape, which is a good estimate shape of the chin, on the snake convergence.

5.3.4 Chin Fitting by Active Contour Model

The energy associated with a balloon snake is recalled here:

$$E = \int_0^1 \frac{\alpha}{2} \left|v'(s)\right|^2 + \frac{\beta}{2} \left|v''(s)\right|^2 + E_{image}(s) + E_{pressure}(s)ds \tag{5.1}$$

Again the snake is in a parametric form. But since the polar system is used, the snake is parametrised as $v(s) = (r(s), \theta(s))$, $s \in [0, 1]$. Where $r(s)$ is the distance from the mouth centre to the control point $v(s)$ and $\theta(s)$ is the angle of the inclination of $v(s)$. The first two terms in Equation 5.1 denote the energy of the regularity of the snake, which controls the elasticity and rigidity of the model. The third term is the potential associated with the image force (F_{image}) derived from the image. The last term is the potential associated with the pressure force ($F_{pressure}$).

The image force is formulated based on the chin characteristic found in Section 5.2. The image force is capable of pushing the control points to an intensity valley. The mathematical form of this term is:

$$F_{image}(v(s)) = \gamma(r, \theta) \times \frac{dI}{dr}(v(s)) \tag{5.2}$$

the image intensity gradient term $\frac{dI}{dr}$ is evaluated at the point $v(s)$ on the snake along the corre-

sponding radial line. The image force (F_{image}) is an expanding one when the image intensity gradient is negative and a shrinking one otherwise. $\gamma(r, \theta)$ is a weighting factor whose value is dependent upon the image topology and an algorithm allowing self- adjustment is proposed in Section 5.3.5.

The pressure force ($F_{pressure}$) is simply a constant inflating force to make the snake progress towards the chin. The pressure force must be set to a value smaller than the image force [32] therefore the snake can be caught by the intensity valley of the chin. In practice, the pressure force is set to a quite small value (as long as it can blow up the snake is sufficient). Equation 5.3 depicts its mathematical form. $\vec{n(s)}$ is the unit vector pointing in the outward direction for each control point $v(s)$ and ρ is a constant positive number equal to $0.5 \times \gamma_{min}$. γ_{min} will be discussed in Section 5.3.5.

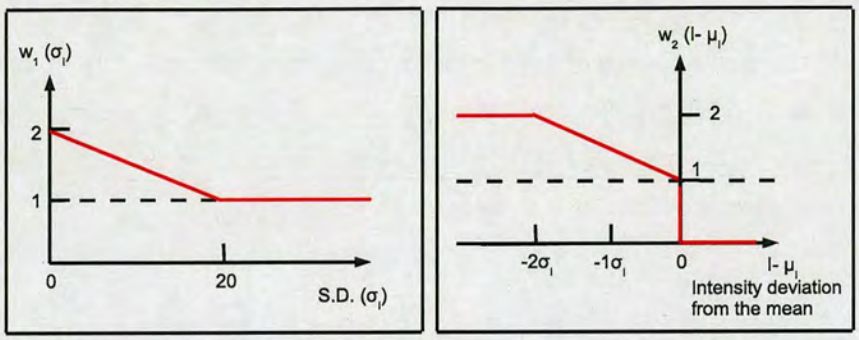
$$F_{pressure}(s) = \rho \times \vec{n(s)} \tag{5.3}$$

The snake, which grows with this inflating force, will find its best fit of the chin by minimising the total energy of Equation 5.1 after a number of iterations (500 iterations are used in the experiment). Alternatively, one may set a "maximum pixel drifting threshold" to flag the reach of the convergence.

5.3.5 Topologically Adaptive Snake

It is obvious that some face images contain more "noise" and "edges" than others. For example, wrinkles, pimples, freckles and stubble increase the variance of the facial skin dramatically. Even worse, the intensity valley of the chin contour varies depending upon facial tissue and bone structure. It has been found that using a set of fixed snake parameters cannot fit chins in a diverse range of facial images. To overcome this problem, a statistical model for intensity is used to automatically adjust the weighting factor $\gamma(r, \theta)$.

Since each control point is only allowed to move radially, a statistical model for intensity can be automatically constructed for each radial line. The pixels on the radial line (but within the search region) are collected and the intensity mean (μ_I) and standard deviation (σ_I) can be obtained. μ_I and σ_I represent the important facial skin dynamic and are used to adjust the $\gamma(r, \theta)$.



(a) piece-wise function of $w_1(\sigma_I)$ **(b)** piece-wise function of $w_2(I - \mu_I)$

Figure 5.7: *Figures show the functions of* $w_1(\sigma_I)$ *and* $w_2(I - \mu_I)$.

Clearly, a radial line containing a higher standard deviation (σ_I) implies a higher "noise" level, which indicates that a smaller weighting factor (γ) should be applied for the image force to avoid the snake being trapped at an undesired "noise" concavity. Conversely, a lower σ_I implies a more homogeneous skin patch suggesting that a larger γ should be applied to boost the chin characteristic.

Additionally, the mean (μ_I) can be used to check for the presence of the chin. A larger γ is used for the control point whose pixel intensity is much lower than μ_I, whereas γ is set to zero if the control point has higher intensity than μ_I. This assumes that the intensity valley of the chin should plunge below the mean intensity of the facial skin.

As a result, $\gamma(r, \theta)$ can be described in Equation 5.4 and two piece-wise functions, one for $w_1(\sigma_I)$ and another for $w_2(I - \mu_I)$, are implemented.

$$\gamma(r, \theta) = w_1(\sigma_I) \times w_2(I - \mu_I) \times \gamma_{min} \tag{5.4}$$

Where $w_1(\sigma_I)$ has the value 2 when σ_I is zero, reducing linearly to 1 when σ_I is > 20. This gives a larger γ for a homogeneous skin region (low σ_I) to emphasise the chin contour valley and a smaller γ as the "feature noise" level (σ_I) increases, in order to avoid the snake being trapped at an undesired "noise" concavity. This function is depicted as Figure 5.7(a). $w_2(I - \mu_I)$ has the value 2 when $(I - \mu_I)$ (the intensity deviation from the mean) is $< -2\sigma_I$, again decreasing linearly to 1 when $I = \mu_I$ to encourage the snake to converge to a "dark" valley

but avoid instability. A larger γ is used when the image intensity at the control point is much lower than the mean μ_I, but w_2 and therefore γ is set to zero if the intensity is higher than the mean. (As the intensity valley of the chin contour is assumed to be below the mean intensity of the facial skin.) This function is depicted as Figure 5.7(b). Thus $\gamma(r, \theta)$ has an overall dynamic range of $4 \times \gamma_{min}$. The γ_{min} (typically 0.02) was determined by extracting the chins from several facial images containing high noise levels.

5.3.6 Results of Chin Fitting by the Initial Approach

This initial approach achieves very high fitting rate for the balanced frontal lit images in XM2VTS database [16]. As stated earlier, the external force now is adaptive, thus the snake can fit the chin contour for various skin colour, gender, age, and with or without stubble (Figure 5.8(a) - Figure 5.8(h)). Faces with heavy beards, which usually result in invisible (or partially invisible) chins, account for most of the false fitting (Figure 5.8(i) and Figure 5.8(j)). Furthermore, this approach can handle the chins well for small head rotation (Figure 5.8(k) and Figure 5.8(l)).

However, this initial approach can not fit the chins under unbalanced lighting conditions, because this adaptive snake uses intensity in finding the chin contour. The shadow and spurious edges caused by the unbalanced lighting can overrun the chin characteristic and hence the chin is not fitted. Figure 5.9 shows some examples.

5.4 Chin Fitting Algorithm for Unbalanced- and Half- Lit Face Images

To overcome the problems caused by the unbalanced lighting, a technique exploiting the symmetrical property of the face and employing a second snake has been developed. It includes a functionality to detect the direction of light so that the bright and dark sides of the face are identified. An initial active contour model, which was introduced in Section 5.3, can be implemented to fit the bright side of the chin. The dark side of the chin which may be corrupted with shadows and spurious edges is extrapolated by using the symmetrical property of the face. This extrapolation is then used as the initialisation of the second snake. Thus the complete chin contour can be found by joining the fitting of the initial snake and the second snake. Figure 5.10 shows the block diagram flow of this new chin fitting approach.

(a) subject 00011 **(b)** subject 00111 **(c)** subject 02711

(d) subject 02321 **(e)** subject 00511 **(f)** subject 03411

(g) subject 03321 **(h)** subject 00911 **(i)** subject 03121

(j) subject 01821 **(k)** subject 01121 **(l)** subject 02511

Figure 5.8: *Results of the chin fitting in frontal-lit images by using the initial fitting algorithm*

(a) Akiyo (b) Susie (c) subject 0011

Figure 5.9: *Results of the chin fitting under unbalanced lighting conditions by using the initial fitting algorithm. Notice that the chin characteristic found in Section 5.2 is covered by the shadow.*

5.4.1 Detecting the Direction of the Light

In real world head-and-shoulders images, balanced, frontal illumination, such as shown in Figure 5.8, may not always occur. In most cases, light can be expected from above the head, but with one side stronger than the other. Consequently, a shadow is formed on the lower part of the poor-lit side of the face. Depending upon the severity of light imbalance, the shadow can be formed in a confined region below the chin (Figure 5.9(a) and Figure 5.9(b)) or cover a wider region of the poor-lit side of the face (Figure 5.9(c)).

A simple technique to detect whether the light is unbalanced and on which side of the face the shadow will form is introduced. First, a rectangular box is drawn below the mouth centre, with the length equal to the distance between the two jaw points and the height equal to twice the distance of the nose - mouth centres, as shown in Figure 5.11. Then, intensities of the pixels on the left and right side of the box (named Region L and Region R in Figure 5.11) are collected separately. Thus the average intensities of Region L and Region R can be computed. If the shadow is formed on one side of the face, it can be expected that the average intensity of the shadow side will be much smaller. In the experiment, a threshold of one side being more than 1.5 times brighter than the other side or the brightness difference being greater than 50 (greyscale $0 \sim 255$) is taken to indicate unbalanced lighting.

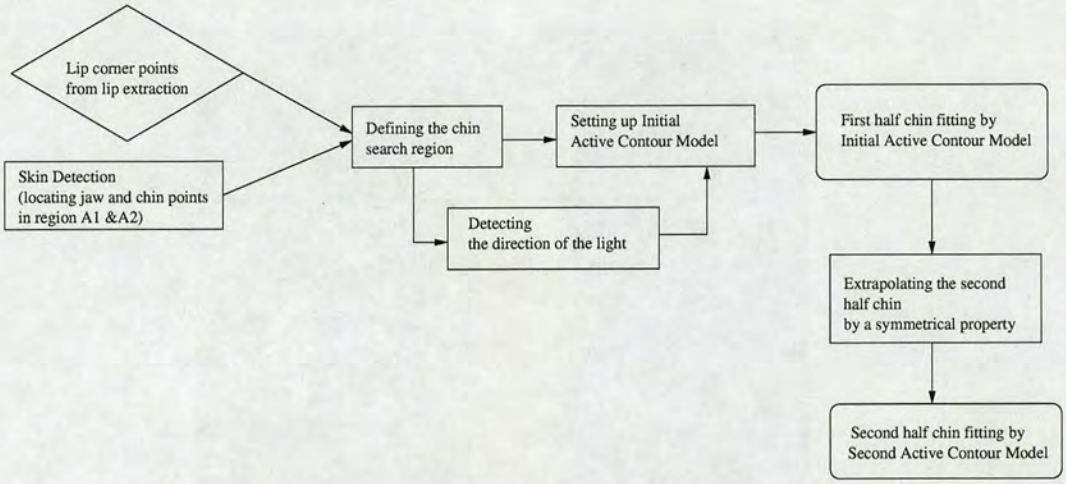


Figure 5.10: *The flow chart illustrates the chin fitting approach under unbalanced lighting conditions*

5.4.2 Chin Extrapolation in the Dark Side of the Face

Once the dark and bright sides of the face are identified, the initial fitting technique described in Section 5.3 is used to fit the bright side of the chin. The dark side of the chin is extrapolated using the symmetrical property of the face.

Figure 5.12 illustrates how the extrapolation is performed. For simplicity, the face is assumed to have no inclination in the x-y plane (that is, the eyes are levelled at a horizontal line) and thus a vertical line through the mouth centre should pass the lowest point of the chin, as shown in Figure 5.12. This assumption can be compensated by exploiting eye or lip corner positions to obtain the true inclination of the face. Since the jaw points (S_0 and S_{n-1}) on both sides of the face have been found, the chin on the shadow side of the face (S_{n-2}, S_{n-3}...) can be extrapolated by using the ratio of two jaw points deviated from the centre vertical line. In other words, the ratio $J_L : J_R = \overline{S_1C_1} : \overline{S_{n-2}C_1} = \overline{S_2C_2} : \overline{S_{n-3}C_2}$ and so on in Figure 5.12. Since the control points on one side of the face are already located by the initial fitting, the control points on another side can be extrapolated using the ratio $J_L : J_R$. The extrapolation using a fixed ratio together with face inclination adjustment can handle the chins for reasonable 3-D head rotation.

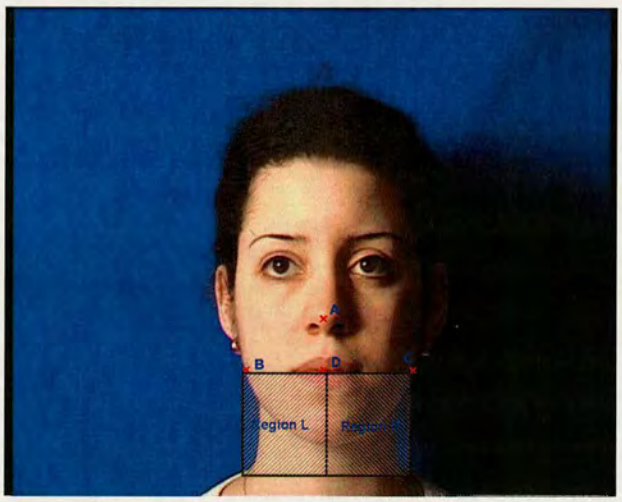


Figure 5.11: *The average intensities are obtained from Region L and Region R to determine the direction of the light and the location of the shadow. Point A, B, C and D are the nose centre, left jaw point, right jaw point and mouth centre, respectively.*

5.4.3 Chin Fitting by the Second Snake

To make the extrapolation of the chin match the local details, a second snake using the extrapolation as its initialisation is employed. The second snake is the same as the initial snake, except that the inflating force is now removed and the external (image) force is modified. Since the extrapolation is now very close to the true chin outline, without the aid of the inflating force, an image force produced by the image is sufficient to attract the snake to the feature. The image force is different from that used in the initial approach. Since the chin is covered by the shadow, the intensity valley previously used for chin extraction is now no longer present. Instead, due to the structure of the protruded chin bone, the chin often shows a rapid intensity change (the chin edge) in the shadow area. This can be a reflection on the chin bone (Figure 5.13(b)) or just

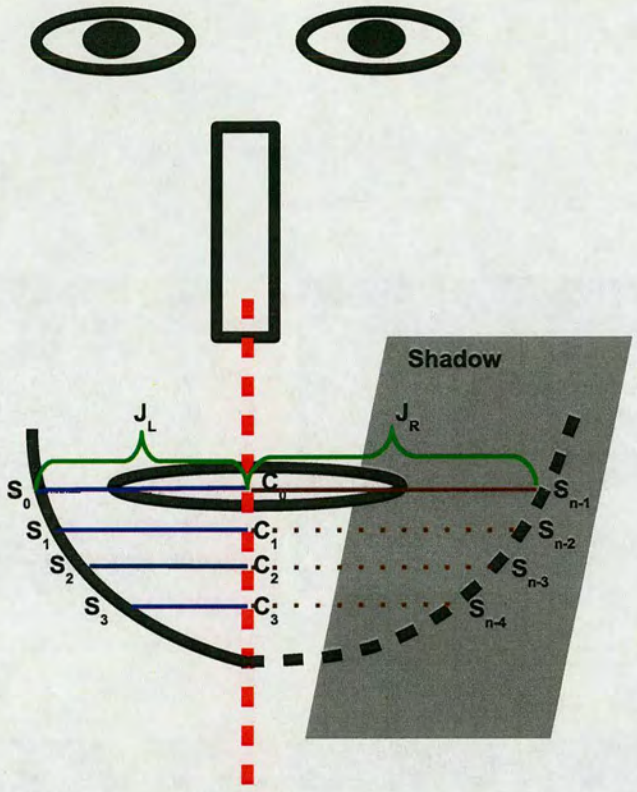


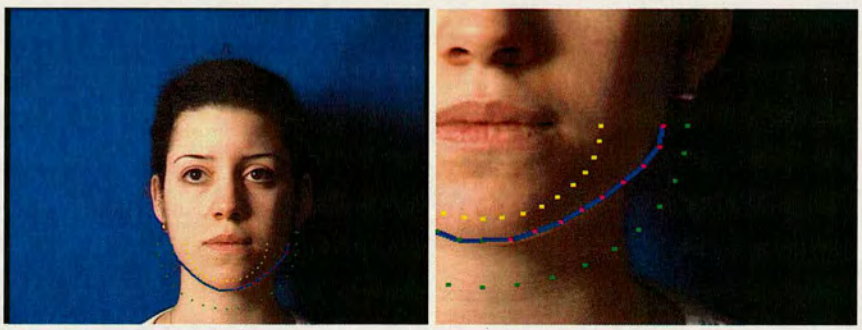
Figure 5.12: *The chin under the shadow is extrapolated by its found counterpart using a constant ratio computed from the two jaw points and the centre vertical line*

because the higher curvature of the chin makes the brightness change more quickly. In addition, in order to obtain a stable convergence, the control points of the second snake are only allowed to move within ± 5 pixels from its initial position.

Figure 5.13 makes a comparison between using an extrapolation only and an additional second snake. The latter achieves better results in terms of matching the local details. Figure 5.13(d) shows better fitting on the chin edge and face boundary.

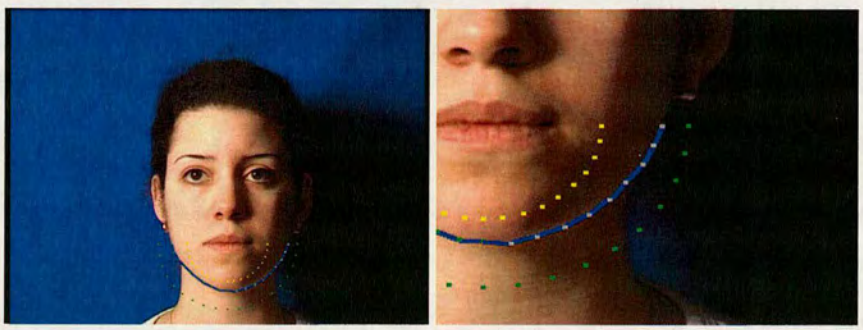
5.4.4 Results of the Chin Fitting by the Improved Approach

Figure 5.14 shows the results of the chin fitting in the half-lit face images in XM2VTSDB [16]. Again, this approach is very successful in fitting the chins regardless of light direction, skin colour, gender, age, and with or without stubble. Failures are mainly due to beards (Figure 5.14(h)) and a "soft" chin (Figure 5.14(i)) which is also hardly discernible to human eyes. To



(a) using chin extrapolation only

(b) enlargement of the shadow part of the chin in Figure 5.13(a)



(c) implementation of the second snake

(d) enlargement of the shadow part of the chin in Figure 5.13(c)

Figure 5.13: *Comparison between the fitting by extrapolation and the further refinement using the second snake. It can be seen clearly that Figure 5.13(d) is better fitted than Figure 5.13(b) on the chin edge and face boundary.*

further test the capability of the algorithm, images from two standard video sequences, Susie and Akiyo, are used. Susie and Akiyo images represent very common unbalanced lighting conditions occurring in head-and-shoulders sequences. In both sequences, sample images are tested for every 10 frames (the images with the chins occluded by the phone in Susie sequence are discarded). The results for Akiyo images are better than for Susie's. It is because Susie images contain larger head rotation and have less defined chin edges (probably due to her make-up). Figure 5.15 shows some examples of the fitting in these images.

5.5 Conclusions and Future Work

In this chapter, two chin fitting approaches have been proposed. The first one employs an "adaptive, inflating balloon, intensity snake" to fit the chin in the frontal-lit images. The snake is initialised as a semi-circle inside the face region and inflated to fit the chin contour. The adaptability allows the snake to self-adjust its parameters according to the image topology so that it can fit the chins in a wide range of face images. However, due to the intensity dependence, this approach is unsuccessful in fitting the chins in half- or unbalanced-lit face images. The second approach exploits the symmetrical property of the face to extrapolate the chin under the shadow. It is then followed by a second snake to refine the fitting to match the local details. These two approaches have been tested in XM2VTSDB [16] and standard video sequences. The first approach is successful in fitting the chins in the frontal-lit XM2VTSDB, while the second one is also successful in the half-lit XM2VTSDB and the video sequences.

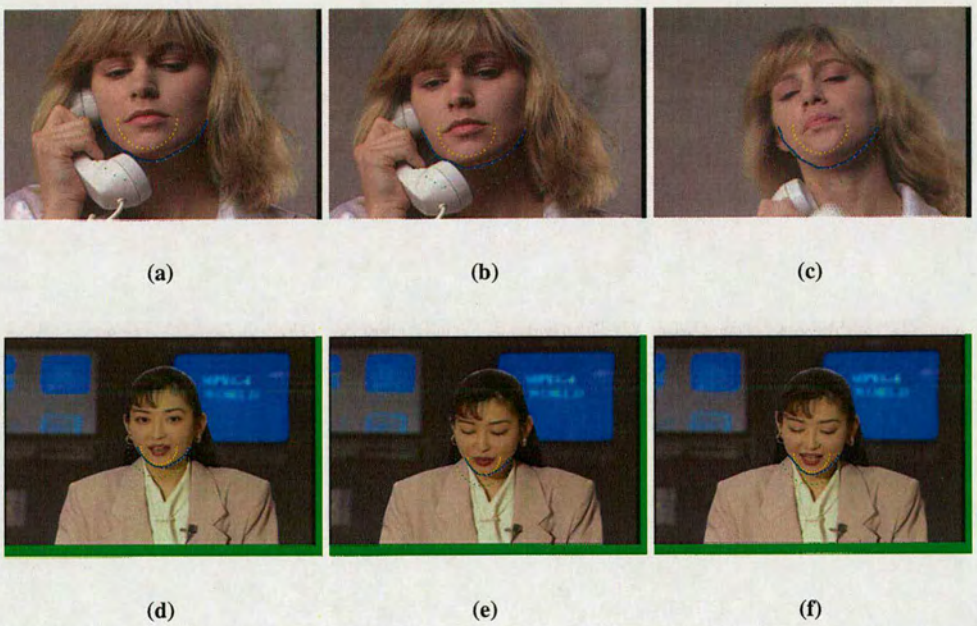
The future work will focus on fitting the partially occluded chins and dealing with large head rotation. To do this, a more effective use of related facial features to estimate the location of the chin is needed. Furthermore, the chin location together with the locations of the eyes, eyebrows, nose and lip will be able to improve the accuracy of the wire-frame face model fitting. This will lead to more realistic expression synthesis and facial animation.

(a) subject 000 **(b)** subject 001 **(c)** subject 027

(d) subject 036 **(e)** subject 005 **(f)** subject 034

(g) subject 009 **(h)** subject 045 **(i)** subject 042

Figure 5.14: *Chin fitting in half-lit XM2VTSDB [16] by the improved approach.*



(a) (b) (c)

(d) (e) (f)

Figure 5.15: *Chin fitting in standard video sequences by the improved approach. Figure 5.15(a) to Figure 5.15(c), fitting in Susie images. Figure 5.15(d) to Figure 5.15(f), fitting in Akiyo images.*

Chapter 6
Model Face Adaptation

6.1 Introduction

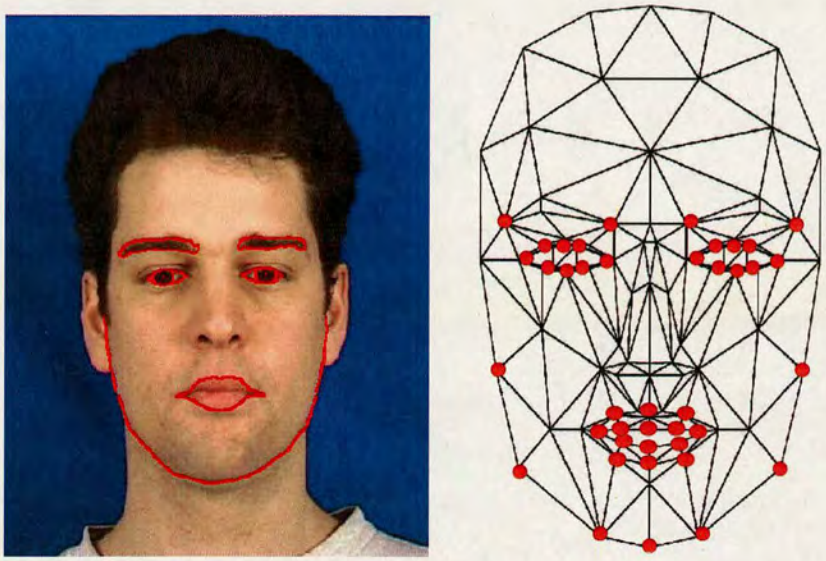
To produce a useful face model, the model must be fitted to the face accurately and must also be deformable. The deformability results in production of diverse facial actions and expressions while the high fitting accuracy ensures these animations occur in the correct places.

In this chapter, two approaches for automatic face model adaptation are presented. Both use the face texture and the knowledge of the formerly fitted feature contours. The differences between these two are the accuracy and usability. The first approach is very accurate but can only be used among the same models. By contrast, the second approach sacrifices accuracy but is compatible with other face models.

Candide-3 [17] is the model for demonstrating fitting and animation. It consists of 85 control parameters (6 global pose parameters, 14 shape units and 65 animation units), 113 vertices and 168 surface polygons. This moderate-to-low complexity allows the concept of fitting and animation to be demonstrated without much deterioration of face detail and realism. The detail of Candide-3 is described in Section 2.4.

6.2 Adaptation with Vertex Position Correction

This approach starts with an initial adaptation of the Candide-3 model to the face. The initial adaptation employs Hillman's [4, 10] or Ahlberg's [13] technique which applies an Active Appearance Model (AAM) to fit the faces. However, the experience from previous research [97] shows that the important feature points need to be fitted extremely accurately in order to generate natural and realistic facial animations. This cannot be achieved solely by using appearance-based approaches; a feature-based measure must be incorporated. Consequently, a "Vertex Position Correction" scheme is developed as a post-processing for the initial model adaptation.



(a) input image with fitted feature contours

(b) corresponding Candide-3 vertex points (iris points are not marked)

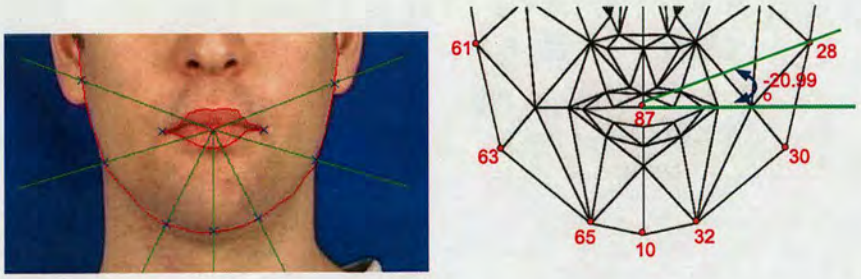
Figure 6.1: *Input image with fitted feature contours and their corresponding vertex points in Candide-3*

A set of 49 important feature points are extracted from the fitted feature contours to correspond to their vertex positions in Candide-3. These points are used to augment the fitting accuracy around the important facial features, which are used to define the face area or are frequently involved in animation. The number (49) is set according to the structure of the Candide-3 model. It is beneficial to have these found feature contours available as more points can be extracted from them if the implementation model contains more vertices and polygons and it is necessary to do so.

The 49 feature points are divided into three groups.

- Group 1: chin contour
- Group 2: eye and eyebrow contour
- Group 3: lip contour

Figure 6.1(a) shows the overall feature contours (lip, eye, eyebrow and chin) as found in previous chapters and Figure 6.1(b) shows the vertices required to be matched in Candide-3 (eight



(a) lower part of the input face with marked chin points

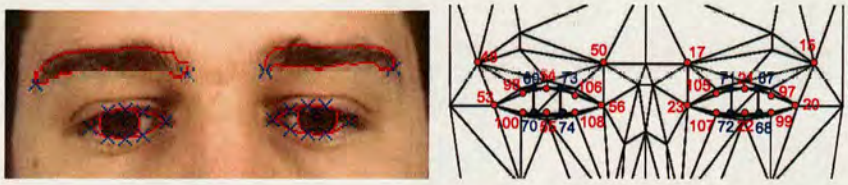
(b) corresponding Candide-3 chin vertex points

Figure 6.2: *Lower part of the input face and the Candide-3 model*

iris vertex points are not marked). In the chin region, 7 points are required to be extracted from the contour to correspond to the Candide chin vertex points 28, 30, 32, 10, 65, 63, and 61 (shown in Figure 6.2(a) and Figure 6.2(b)). A radial search method is used to find the correspondence. Assuming the face to be fitted is in the "neutral state" [1], the chin vertex points in Candide-3 subtend the same set of constant angles with respect to the mouth centre as the chin points do in the input face image. More specifically, if a horizontal line is drawn through the mouth centre, the angles subtended by vertices 28, 30, 32, 10, 65, 63 and 61 are -20.99^o, 16.48^o, 63.43^o, 90.0^o, 116.57^o, 163.52^o and 200.99^o (see Figure 6.2(b) for an example). These angles are calculated in the image Cartesian system (the origin is at the top left corner of the image; x-coordinate increments rightwards and y-coordinate increments downwards.) and the Candide-3 original vertex positions are used. Thus, to locate the chin points in the input image, the mouth centre is identified first and 7 radial lines are drawn from the mouth centre, with the calculated angles, to obtain the intercept points on the chin contour. The mouth centre is located at half way between the lip corners. The found chin points for Figure 6.1(a) are marked in Figure 6.2(a). These chin points provide additional information about the face boundary (such as the head width and jaw position) which is often falsely fitted by using solely the appearance-based approach [4].

There are 28 feature points to be fitted in the eye and eyebrow region (see Figure 6.1(b)). These are the corners of the eyebrows (4 points), the corners of the eyes (4 points) and 8 points (4 points for each iris) to form 2 squares enclosing the irises. The other 12 points are not strictly

[1] see [143] for the neutral face definition



(a) eye and eyebrow region of the input face with marked feature points (iris points are not marked)

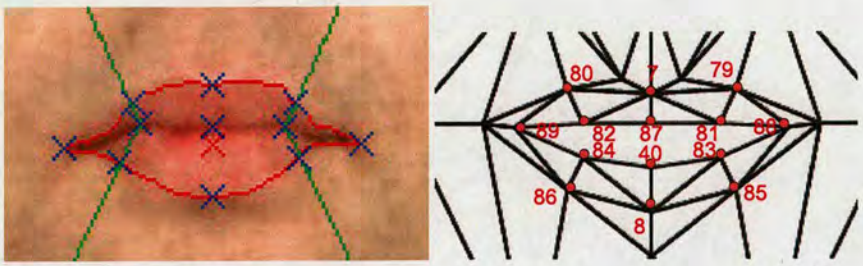
(b) corresponding Candide-3 eye and eyebrow vertex points

Figure 6.3: *Eye and eyebrow region of the input face and the Candide-3 model*

defined in the Candide-3 model. Each upper eyelid is represented by 3 points (6 points for two eyes) and so is the lower eyelid (6 points for two eyes). Thus these make up the total number of the feature points to 28.

The corner points of the eye and eyebrow can be located in the input face by identifying the extremities of the found eye and eyebrow contours (see Figure 6.3(a)). The less defined eyelid points are located by quartering the horizontal distance between the eye corners on the found upper and lower eyelid contours. The iris is enclosed by a square. The locations of the square vertices (vertices 69, 70, 73 and 74 for the left eye and vertices 71, 72, 67 and 68 for the right eye) are calculated from the found iris centres and radii. The eyelid and iris points are important in avoiding the fitting corruption of the eye region which is common in appearance-based approaches as eye region texture is complex and noisy, resulting in a high mismatch rate. Figure 6.3(a) shows the marked eye and eyebrow feature points for Figure 6.1(a) (the iris points are not marked).

There are 14 important feature points selected in the mouth region (see Figure 6.4(b)). These include two inner lip corners (vertices 89 and 88), 6 points on the inner lip boundary (vertices 82, 87, 81, 84, 40 and 83) and 6 points on the outer lip boundary (vertices 80, 7, 79, 85, 8 and 86). Notice the tips of Cupid's bow of the upper lip are not used as such a characteristic does not always appear in lips. To mark these feature points in the input face, the corners are located by identifying the extremities of the lip contour (see Figure 6.4(a)). The 6 inner lip boundary points merge to only 3 points because the mouth is closed in the neutral state. Thus vertex 82 overlaps vertex 84, vertex 87 overlaps vertex 40 and vertex 81 overlaps vertex 83 respectively. These inner boundary points must be situated on the lip contact line which is



(a) mouth region of the input face with marked feature points

(b) corresponding Candide-3 mouth vertex points

Figure 6.4: *Mouth region of the input face and the Candide-3 model*

the low intensity valley connecting the lip corners (as the green lines shown in Figure 3.9 and Figure 3.10 in Chapter 3). They also quarter the distance between the lip corners. The outer lip boundary points are located by using the geometric relationships between themselves and the inner boundary points. Vertices 7 and 8 are on the same vertical line through vertex 87. Vertices 79, 80, 85 and 86 are located by applying a similar radial search method to the chin points, which uses vertices 81, 82, 83 and 84 as the centres. As a result, vertices 79, 80, 85 and 86 are found as the intercept points of the green lines and the lip contour in Figure 6.4(a). These mouth points improve the accuracy of the fitting of the mouth region when they are animated to produce expressions and speech.

After the set of 49 important feature points are all matched to their Candide vertex positions, the initially adapted model (by using an appearance-based approach [4, 10, 13]) can be refined. First these 49 vertex points on the adapted model are relocated to the new locations found on the input face. Second, in order to make the vertex points distribute evenly, the neighbouring points of these relocated vertices are also adjusted. The adjustment is performed by proportionally moving these points with respect to the "reference objects". In the mouth region, the mouth width is used as the reference and the neighbouring points to be adjusted are vertices 9, 64 and 31 (marked as blue circles in Figure 6.5). The mouth width is set as the distance between the inner lip corners (i.e. the distance between vertices 89 and 88). In the original Candide-3 model, the distance between vertex 89 and vertex 88 ($\overline{V_{89}V_{88}}$) is 0.400 and the distances between vertex 9 and vertex 8 ($\overline{V_9V_8}$), vertex 64 and vertex 89 ($\overline{V_{64}V_{89}}$) and vertex 31 and vertex 88 ($\overline{V_{31}V_{88}}$) are 0.065, 0.046 and 0.046, respectively. Thus the ratio of each distance w.r.t. the mouth width is obtained as $\frac{0.065}{0.400}$, $\frac{0.046}{0.400}$ and $\frac{0.046}{0.400}$. In consequence, the vertices 9, 64 and 31 can be moved

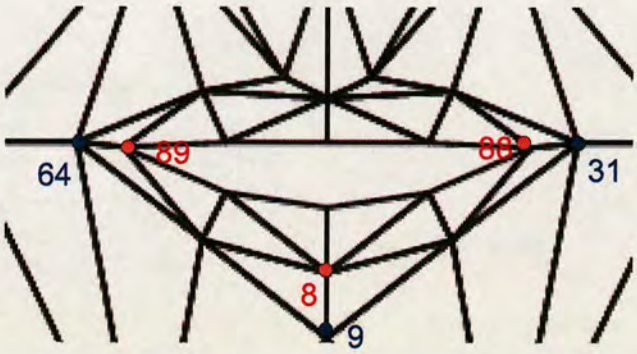


Figure 6.5: *The neighbouring points (marked as blue circles) to be adjusted in the mouth region.*

using these ratios w.r.t. the true mouth width in the input face.

In the eye region, the eye width is used as the reference to adjust the locations of the neighbouring vertices. Take the "left eye" (from the reader's point of view) as an example (Figure 6.6), the eye width is set as the distance between vertex 53 and vertex 56 and the neighbouring points to be adjusted are vertices 96, 102, 52, 57, 104 and 110 (marked as blue circles in Figure 6.6). In the original Candide-3 model, the distance between vertex 53 and vertex 56 ($\overline{V_{53}V_{56}}$) is 0.340 and the distances between vertex 96 and vertex 98 ($\overline{V_{96}V_{98}}$), vertex 102 and vertex 100 ($\overline{V_{102}V_{100}}$), vertex 52 and vertex 54 ($\overline{V_{52}V_{54}}$), vertex 57 and vertex 55 ($\overline{V_{57}V_{55}}$), vertex 104 and vertex 106 ($\overline{V_{104}V_{106}}$) and vertex 110 and vertex 108 ($\overline{V_{110}V_{108}}$) are 0.015, 0.009, 0.021, 0.018, 0.015 and 0.009, respectively. Thus the ratio of each distance w.r.t. the eye width can be computed. In consequence, vertices 96, 102, 52, 57, 104 and 110 can be moved using these computed ratios w.r.t. the true eye width in the input face. The vertices in the right eye region can also be adjusted in the same manner.

Figures 6.7(a) - 6.10(a) show the initial adaptation of the model while Figures 6.7(b) - 6.10(b) show the results of applying the vertex position correction. It can be clearly seen that Figures 6.7(b) - 6.10(b) have significant improvements over Figures 6.7(a) - 6.10(a) in fitting of the chin, eyes, eyebrows and lip. Notice that the nose positions are not corrected using this scheme as it is found that the nose region is fitted well using the appearance-based approach because the texture of the nose is more consistent and distinctive from other parts of the face.

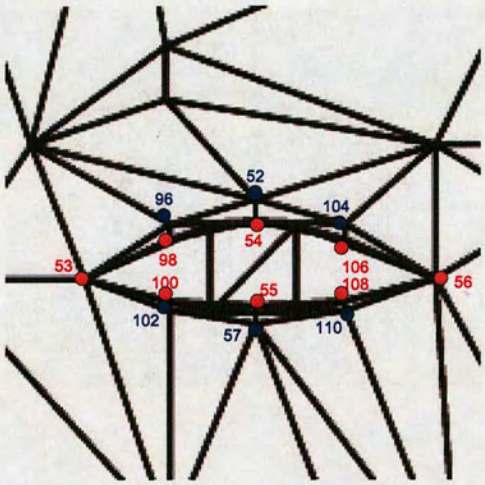


Figure 6.6: *The neighbouring points (marked as blue circles) to be adjusted in the left eye region.*

(a) appearance-fit of subject 00021 **(b)** vertex position correction of Figure 6.7(a)

Figure 6.7: *Comparison between the appearance-only-fit and the post-processed vertex position correction for Subject 00021*

(a) appearance-fit of subject 00111 **(b)** vertex position correction of Figure 6.8(a)

Figure 6.8: *Comparison between the appearance-only-fit and the post-processed vertex position correction for Subject 00111*

(a) appearance-fit of subject 00521 **(b)** vertex position correction of Figure 6.9(a)

Figure 6.9: *Comparison between the appearance-only-fit and the post-processed vertex position correction for Subject 00521*

(a) appearance-fit of subject 02311 **(b)** vertex position correction of Figure 6.10(a)

Figure 6.10: *Comparison between the appearance-only-fit and the post-processed vertex position correction for Subject 02311*

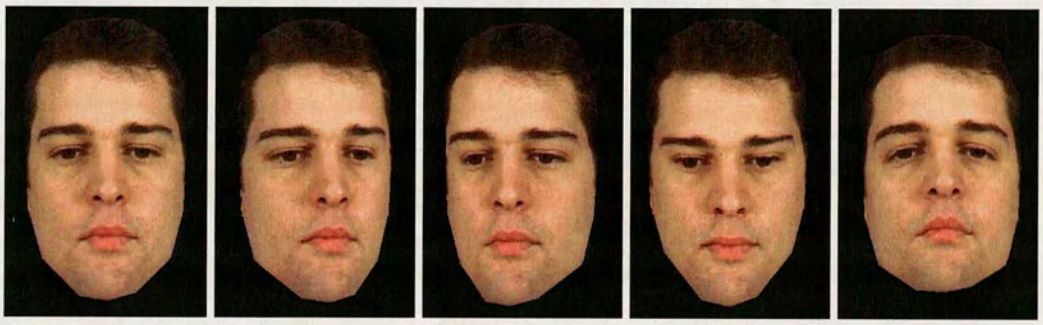
6.2.1 Assessment of the Model Fitting by Animation

One of the best ways to assess the fitting of the model is to animate it. In this section, the model is rotated to different views and given several expressions to perform using the model-based technology (i.e. changing the control parameters). In the rotation test the model is applied with $\pm 15^o$ angle about x and y axes and in the expression test, five major expressions are generated according to the MPEG-4 code book [143].

Figures 6.11, 6.12, 6.13 and 6.14 show the 15^o rotation of the fitted models about x and y axes. They look fairly realistic. Experiments have shown that good realistic side-views can be generated up to about 30^o. Angles greater than this will produce some unrealistic artefacts, particularly noticeable at the nose and nose ridge. This is because no depth information is available from a single fitted frontal face. Notice that Figure 6.14(c) and 6.14(e) show a void texture near the nose. This is due to the bad wireframe fit near the nose so no texture is sampled, as shown in Figure 6.10(b).

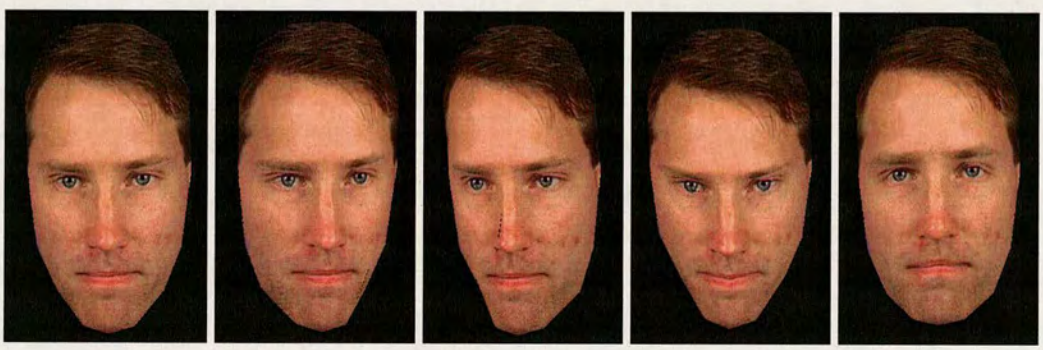
The models are also animated to produce five major expressions – "anger", "fear", "joy", "sadness" and "surprise". These expressions are by far the most distinguishable and reasonably defined. [143] lists major characteristics of each expression defined in the MPEG-4 code book[2].

[2]Actually there are six expressions defined in MPEG-4. However, the sixth expression, "Disgust", has a different definition from another major facial expression group, Paul Ekman et al. [3]. Therefore it is excluded in the test.



(a) original face | **(b)** $+15^o$ y rotation | **(c)** -15^o y rotation | **(d)** $+15^o$ x rotation | **(e)** -15^o x rotation

Figure 6.11: $\pm 15^o$ *rotation about x and y axes for subject 00021*

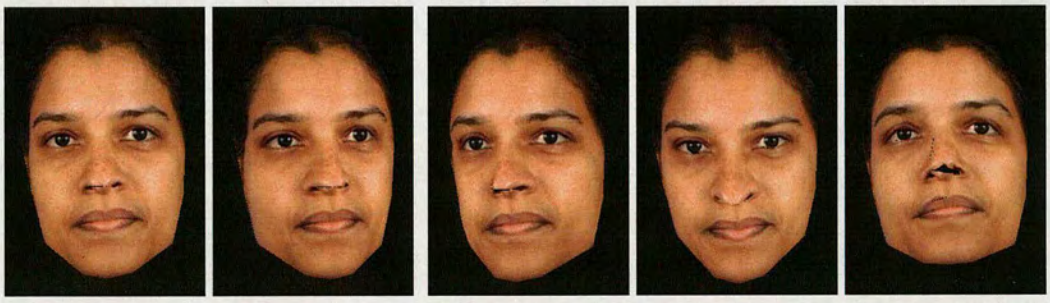


(a) original face | **(b)** $+15^o$ y rotation | **(c)** -15^o y rotation | **(d)** $+15^o$ x rotation | **(e)** -15^o x rotation

Figure 6.12: $\pm 15^o$ *rotation about x and y axes for subject 00111*

(a) original face **(b)** $+15^o$ y rotation **(c)** -15^o y rotation **(d)** $+15^o$ x rotation **(e)** -15^o x rotation

Figure 6.13: $\pm 15^o$ *rotation about x and y axes for subject 00521*



(a) original face **(b)** $+15^o$ y rotation **(c)** -15^o y rotation **(d)** $+15^o$ x rotation **(e)** -15^o x rotation

Figure 6.14: $\pm 15^o$ *rotation about x and y axes for subject 02311*

- Anger: The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or open to expose the teeth.
- Fear: The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.
- Joy: The eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears.
- Sadness: The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.
- Surprise: The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is open.

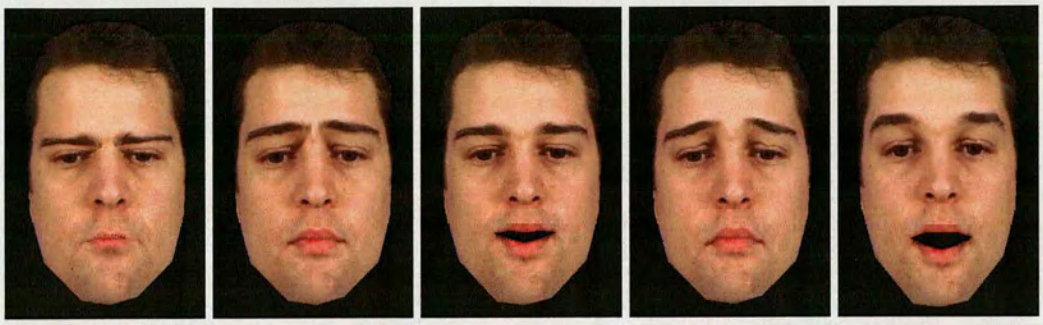
As can be seen, although each expression is textually described in MPEG-4, the exact execution, such as proportion of the facial actions and the "adjectives" (tense, alert and relaxed), remains ambiguous and is left to users' choice. To produce a realistic expression, the values of the animation units have to be determined experimentally and assessed subjectively. A list of Candide-3 Animation Units (AUs) and the interpretation is summarised in Appendix C.2.

Anger is produced by adjusting Animation Unit (AU) 3 (brow lowerer) , 9 (lip presser) and 6 (eyes closed). Notice that AU 6 is assigned an negative value to open the eyes (see Figures 6.15(a), 6.16(a), 6.17(a) and 6.18(a)).

Fear is produced by adjusting Animation Unit (AU) 3 (brow lowered), 39 (raise left inner eyebrow), 40 (raise right inner eyebrow), 45 (squeeze left eyebrow), 46 (squeeze right eyebrow) and 6 (eye closed). Notice that AU 3 and AU 6 are assigned negative values to raise the eyebrows as well as produce the tense and alert eyes (see Figures 6.15(b), 6.16(b), 6.17(b) and 6.18(b)).

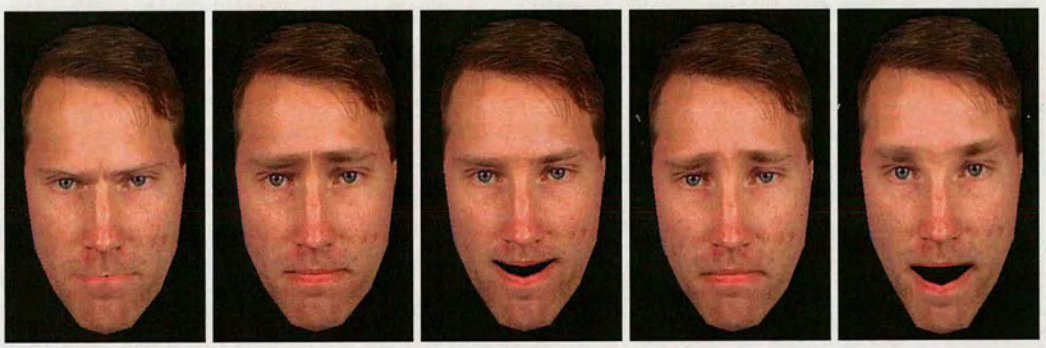
Joy is produced by adjusting Animation Unit (AU) 4 (lip corner depressor), 0 (upper lip raiser), and 1 (jaw drop). AU 4 is assigned a negative value to bend lip corners upward. AU 0 and AU 1 activate a mouth open. AUs controlling eyebrows remain zero to give the relaxed eyebrows (see Figures 6.15(c), 6.16(c), 6.17(c) and 6.18(c)).

Sadness is produced by adjusting Animation Unit (AU) 39 (raise left inner eyebrow), 40 (raise right inner eyebrow), 6 (eyes closed) and 4 (lip corner depressor). All these AUs are assigned



(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.15: *Animation of five major expressions on subject 00021*



(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.16: *Animation of five major expressions on subject 00111*

positive values to produce a sad expression (see Figures 6.15(d), 6.16(d), 6.17(d) and 6.18(d))[3].

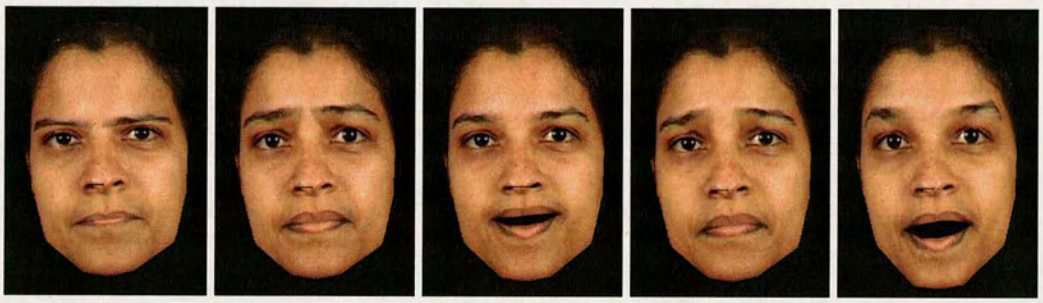
Surprise is produced by adjusting Animation Unit (AU) 3 (brow lowerer), 10 (upper lid raiser), and 1 (jaw drop). Again AU 3 is assigned a negative value to raise the eyebrows (see Figures 6.15(e), 6.16(e), 6.17(e) and 6.18(e)).

From these animations, the model fitting with vertex position correction can be seen to be successful.

[3] AU 4 (lip corner depressor) is added by the author to produce a more realistic sad face

(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.17: *Animation of five major expressions on subject 00521*



(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.18: *Animation of five major expressions on subject 02311*

6.3 Adaptation by Adjusting Model Control Parameters

The second automatic adaptation method is to adjust the control parameters so that the Candide-3 model is reshaped and matched to the face. There are three types of control parameters for adapting the Candide-3 Model - the global pose, shape and animation parameters. However, since the faces in the XM2VTS [16] image database are assumed to be in the "neutral state" (this is nearly true as most of the faces are fairly expressionless), only the global pose parameters and shape parameters are used. The animation parameters will be used for adaptation in the later stage for images containing definite expressions [157]. To improve the fitting at the important feature points, the previously found 49 feature points are used in conjunction with the appearance model to compute the best set of the global pose and shape parameters. This is achieved by an optimisation algorithm which minimises both the appearance error and feature error.

6.3.1 The Methodology

The adaptation algorithm modifies Ahlberg's approach [13]. Using the terminology in [13] and assuming that all the animation units are zero, the Candide-3 Model is parametrised according to a parameter vector

$$\boldsymbol{p} = [r_x, r_y, r_z, s, t_x, t_y, \boldsymbol{\sigma}^T] \tag{6.1}$$

where $r_x, r_y, r_z, s, t_x, t_y$ are global pose parameters. r_x, r_y and r_z are 3D rotations about the x, y and z axis respectively, s is the scale, t_x and t_y are the 2D translations, and $\boldsymbol{\sigma}$ is the shape unit vector (consisting of 14 parameters).

A PCA "face-space" can be formed by fitting Candide-3 manually to several faces and using the fit to reshape the faces back to the standard Candide shape $\overline{\boldsymbol{s}}$. These images differ only in texture - the feature positions are identical in all images. The training set of images $\boldsymbol{S}$ is used to find a PCA transform $\boldsymbol{X}^T$ into the face-space (details see [13]).

Given a parameter set $\boldsymbol{p}$ and an input image $\boldsymbol{i}$, the face in $\boldsymbol{i}$ is reshaped back to the standard face $\overline{\boldsymbol{s}}$ to give a normalised face $\boldsymbol{j}(\boldsymbol{i}, \boldsymbol{p})$. $\boldsymbol{j}$ is mapped into the face-space and back into intensity space to give a reconstructed image

$$\boldsymbol{x}(\boldsymbol{i},\boldsymbol{p}) = \overline{\boldsymbol{x}} + \boldsymbol{X}\boldsymbol{X}^T(\boldsymbol{j}(\boldsymbol{i},\boldsymbol{p}) - \overline{\boldsymbol{x}}) \tag{6.2}$$

where $\overline{\boldsymbol{x}}$ is the mean of the training set $\boldsymbol{S}$ and $\boldsymbol{X}^T$ and $\boldsymbol{X}$ are the forward and reverse PCA transforms respectively. The residual image, $\boldsymbol{r}(\boldsymbol{i},\boldsymbol{p}) = \boldsymbol{j}(\boldsymbol{i},\boldsymbol{p}) - \boldsymbol{x}(\boldsymbol{i},\boldsymbol{p})$ is used to derive an *Appearance Error Measure*

$$e(\boldsymbol{p}) = ||\boldsymbol{r}(\boldsymbol{i},\boldsymbol{p})|| \tag{6.3}$$

Since the training space is constructed solely from well-fitted faces, the PCA space will be able to provide a good reconstruction of well-fitted faces. Thus, the reconstructed image will be similar to the original image and the error is small. Poorly fitted images will be significantly different from the training set and hence will not be well reconstructed, resulting in a large error.

The best fitting parameters $\boldsymbol{p}'$ for the input face image $\boldsymbol{i}$ can be found by finding the parameter set which minimises the error e. Given an approximate fit at the beginning, the parameters are updated by using an update matrix $\boldsymbol{U}$ which maps the residual image to an update vector $\Delta\boldsymbol{p}$ [13]. The process iterates until $e(\boldsymbol{p})$ is small. The initial fit of the model is important. If it is too far from the optimum fit, the optimum may not be reached and the final fit will be found at a local minimum. To solve this, a good initial estimation of the face based on the locations of the important facial features is developed. Three global pose parameters, s, t_x and t_y, are estimated by using 10 out of the 49 found feature positions (4 eyebrow corners, 4 eye corners and 2 lip corners). That is, assuming the input face is the standard Candide-3 shape and is looking forward. Giving 20 parameters (10 feature points with x, and y positions) to solve 3 unknowns, the system is over-specified and can be solved using "*Singular Value Decomposition*".

The main novelty of the proposed algorithm over Ahlberg's approach [13] is that the minimisation error is combined with the appearance and feature measures. Let $\boldsymbol{F}$ be a matrix whose rows are the (x, y) co-ordinates of the found 49 feature points and $\boldsymbol{V_p}$ be a matrix whose rows are positions of the corresponding Candide-3 vertices when adapted using the parameter vector $\boldsymbol{p}$ and projected to 2D. The difference between these two matrices gives a *Feature Error Measure*

$$\epsilon(\boldsymbol{p}) = ||\boldsymbol{V_p} - \boldsymbol{F}||^2 \tag{6.4}$$

which is the difference between the located feature points in the image and the positions of the features given using $\boldsymbol{p}$. The two error measures, e (the appearance error measure) from Equation 6.3 and ϵ (the feature error measure) are combined:

$$\boldsymbol{p}' = \arg\min_{\boldsymbol{p}}(e(\boldsymbol{p}) + \lambda\epsilon(\boldsymbol{p})) \tag{6.5}$$

where λ is a weighting factor able to lift the feature measure to roughly the same scale as the appearance measure. λ is dependent upon the size of the image and the number of the selected points, therefore it is determined experimentally. Figure 6.19 shows the fitting results using this combined appearance and feature measure approach. The results show that the fitting at the feature points has improved massively compared to Ahlberg's approach [13]. Although these fits are slightly less accurate than using "Vertex Position Correction" scheme, they are more valuable as the global pose parameters and shape parameters are identified.

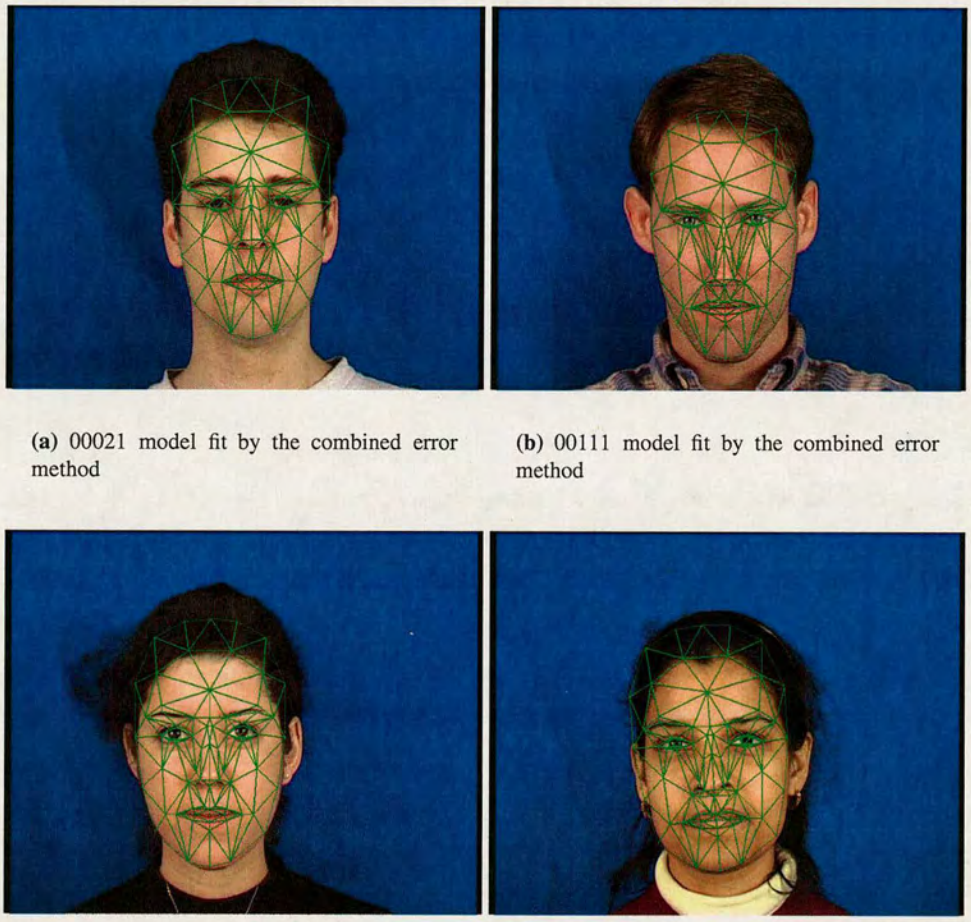
6.3.2 Assessment of the Model Fitting by Animation

The fitted models are again assessed by animation. Three types of animations are used. The first and second are assessed by producing rotation views and artificial expressions, as seen in the Section 6.2.1. The third assessment applies "expression duplication". Definite expressions are captured from the source images [157] and converted to animation units, and "transferred" to the fitted face models.

Figures 6.20 - 6.23 show the 15^o rotation about x and y axes for the fitted models. Again, these views look realistic as long as the angles are kept less than 30^o. Figure 6.23 shows a small area of the background has been included in the model, indicating the global pose parameters are not quite right. Some defects of fitting can also be seen near the nose region of Figure 6.23.

Figures 6.24 - 6.27 show animations of the five major expressions on the fitted models by following the MPEG-4 code book. Since the feature points are not placed at the perfect locations (since the optimisation is driven by minimising both the feature and appearance errors), the fits are slightly less accurate than using the "Vertex Position Correction" scheme. This is most noticeable at the mouth corners of Figure 6.24.

The last animation test is to duplicate the expressions from other images. Five subjects showing different expressions [157] are used as the source images (see Figures 6.28 - 6.32). To extract



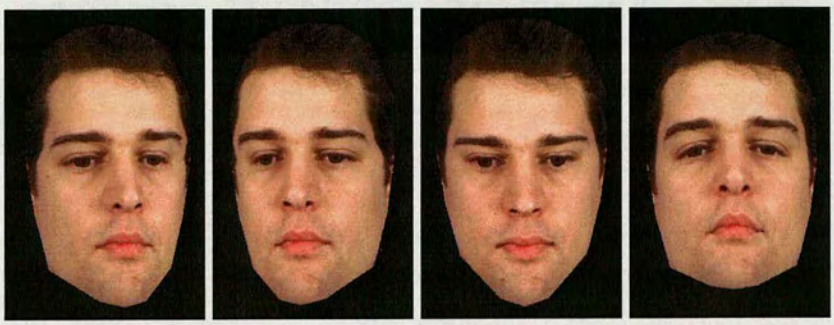
(a) 00021 model fit by the combined error method

(b) 00111 model fit by the combined error method

(c) 00521 model fit by the combined error method

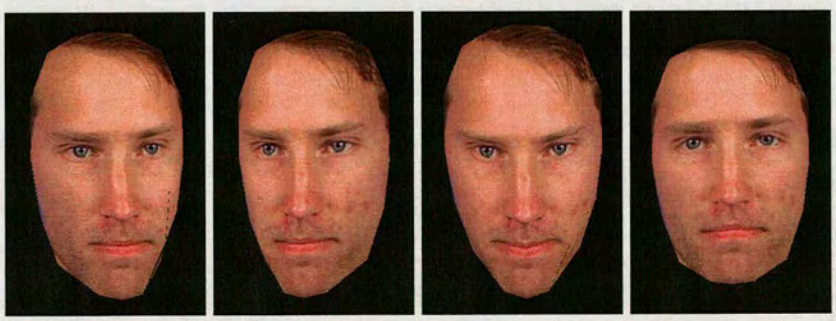
(d) 02311 model fit by the combined error method

Figure 6.19: *Results of the model fits by using the combined appearance and feature measure approach*



(a) $+15^o$ y rotation | **(b)** -15^o y rotation | **(c)** $+15^o$ x rotation | **(d)** -15^o x rotation

Figure 6.20: $\pm 15^o$ *rotation about x and y axes for subject 00021*



(a) $+15^o$ y rotation | **(b)** -15^o y rotation | **(c)** $+15^o$ x rotation | **(d)** -15^o x rotation

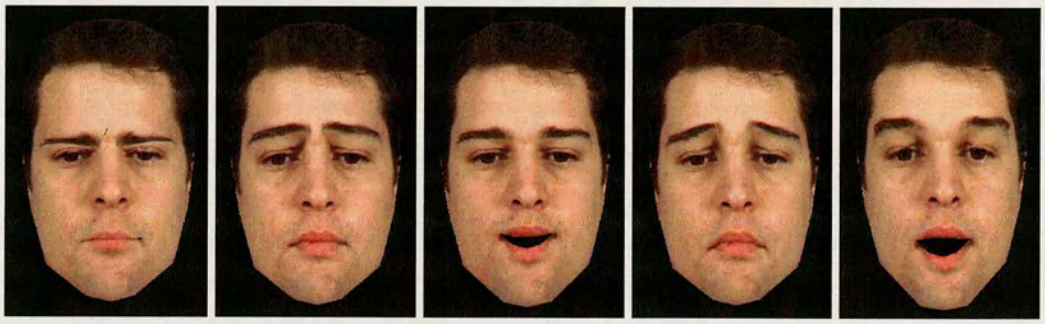
Figure 6.21: $\pm 15^o$ *rotation about x and y axes for subject 00111*

(a) $+15^o$ y rotation **(b)** -15^o y rotation **(c)** $+15^o$ x rotation **(d)** -15^o x rotation

Figure 6.22: $\pm 15^o$ *rotation about x and y axes for subject 00521*

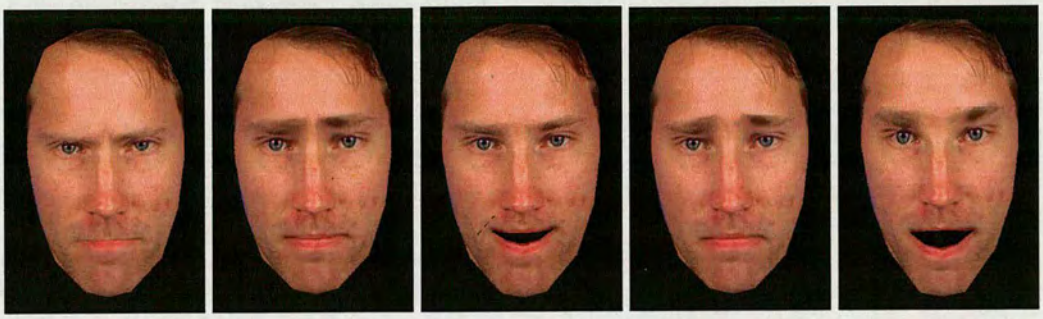
(a) $+15^o$ y rotation **(b)** -15^o y rotation **(c)** $+15^o$ x rotation **(d)** -15^o x rotation

Figure 6.23: $\pm 15^o$ *rotation about x and y axes for subject 02311*



(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.24: *Animation of five major expressions on subject 00021*



(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.25: *Animation of five major expressions on subject 00111*

(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.26: *Animation of five major expressions on subject 00521*

(a) anger **(b)** fear **(c)** joy **(d)** sadness **(e)** surprise

Figure 6.27: *Animation of five major expressions on subject 02311*

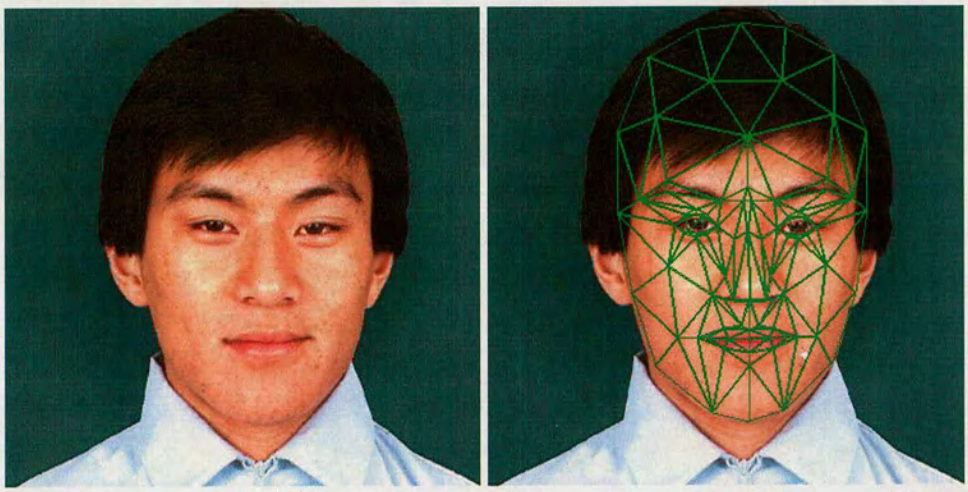
the animation parameters from each expression, the neutral face and expressive face need to be fitted. For fitting the neutral faces (Figures 6.28(a)- 6.32(a)), the method described in this section is used. However, for fitting the expressive faces (Figures 6.28(c) - 6.32(c)), the animation units rather than the shape units are updated. It is assumed that the shape of the subject's face does not change, thus the shape parameters are fixed to the values found in fitting the neutral face and allowing the global pose and animation parameters to be updated. In consequence, the optimiser will compute the best sets of the global pose parameters and animation parameters for the expressive face. The set of the computed animation units can then be implemented in the fitted model faces and generate a so-called "personalised" expression. The "personalisation" means the expression now is no longer the generic one which uses MPEG-4 rules, but is one duplicating a particular person's specific expression.

The "transferability" and "personalisation" are actually the advantages over the "Vertex Position Correction" scheme. Since the "Vertex Position Correction" scheme cannot identify the shape/animation parameters, it is not possible to transfer or personalise the shape/expression from one model to the other. Figures 6.33 - 6.36 illustrate the results of expression duplication from Figures 6.28 - 6.32 on the previously model fitted faces.

6.4 Conclusions and Future Work

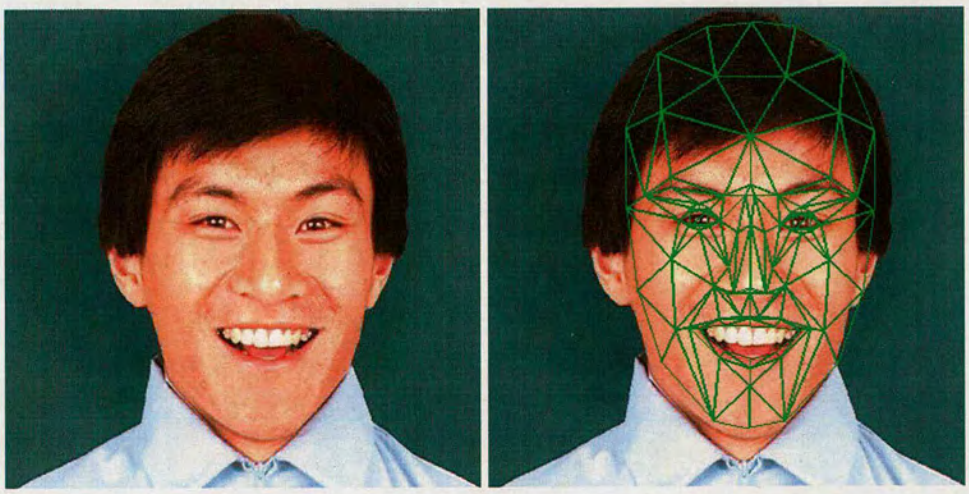
In this chapter, two automatic model adaptation approaches to human faces are presented. The first approach uses Hillman's [4, 10] or Ahlberg's [13] model adaptation, but adds an extra post-processing scheme of "Vertex Position Correction". This scheme is able to adjust the model vertices which correspond to the important facial features to more precise locations. The locations of important features are found in the fitted feature contours described in the former chapters. The corresponding vertices are thus forced to move to the found feature locations and the neighbouring vertices are also moved proportionally. The resultant model fits have been tested by rotation and expression animations. Both tests show this automatic model adaptation is successful.

The second approach is to incorporate a feature measure into an appearance adaptation approach [13]. As a result, the model is fitted to a face when the combined appearance and feature error is minimised. Although it is not as precise as the first approach at the feature locations, it corrects the drawback of Ahlberg's approach [13] that the fitting is usually quite poor



(a) neutral face of subject 01

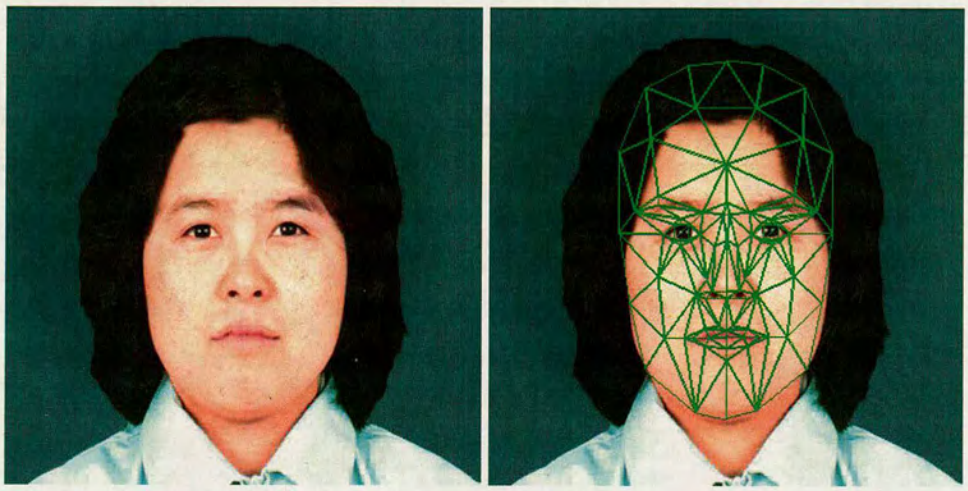
(b) model fit of the neutral face



(c) joyful expression of subject 01

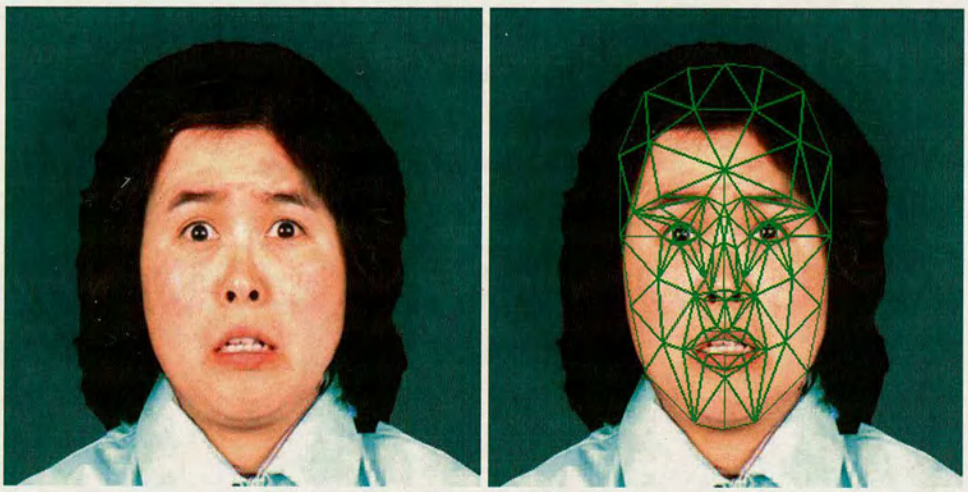
(d) model fit of joyful expression

Figure 6.28: *Model fit of the neutral and joyful face*



(a) neutral face of subject 05

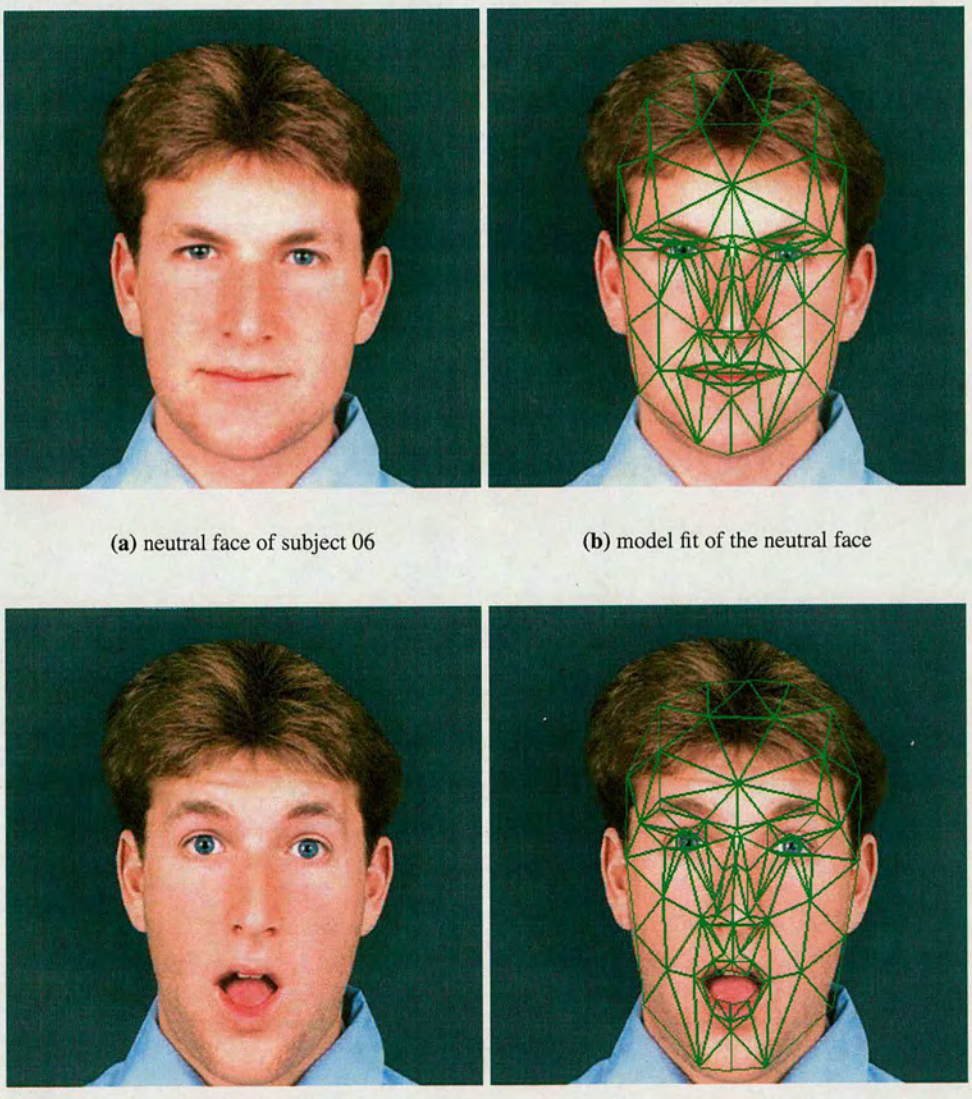
(b) model fit of the neutral face



(c) fearful expression of subject 05

(d) model fit of fearful expression

Figure 6.29: *Model fit of the neutral and fearful face*



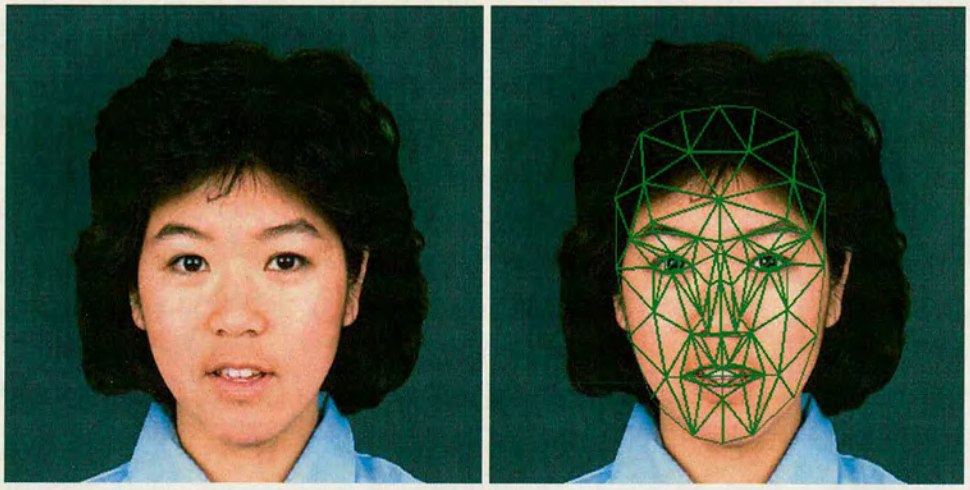
(a) neutral face of subject 06

(b) model fit of the neutral face

(c) surprising expression of subject 06

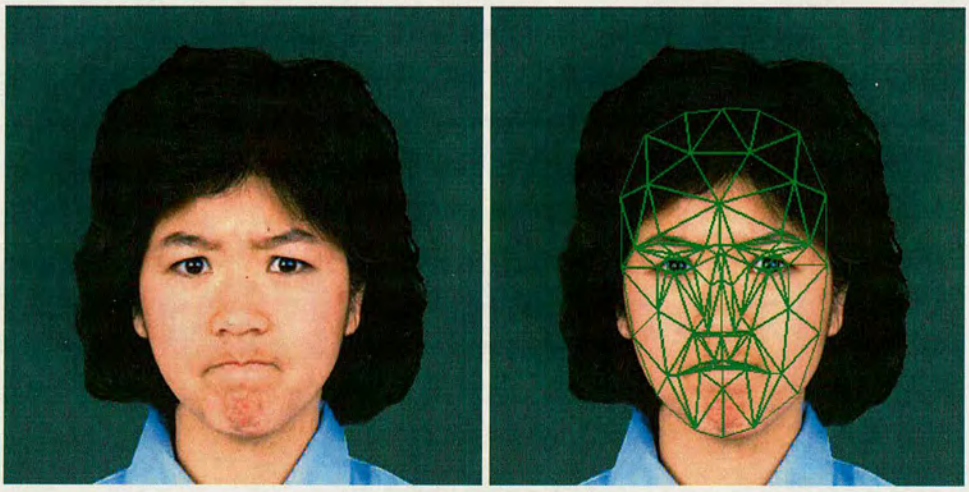
(d) model fit of surprising expression

Figure 6.30: *Model fit of the neutral and surprising face*



(a) neutral face of subject 08

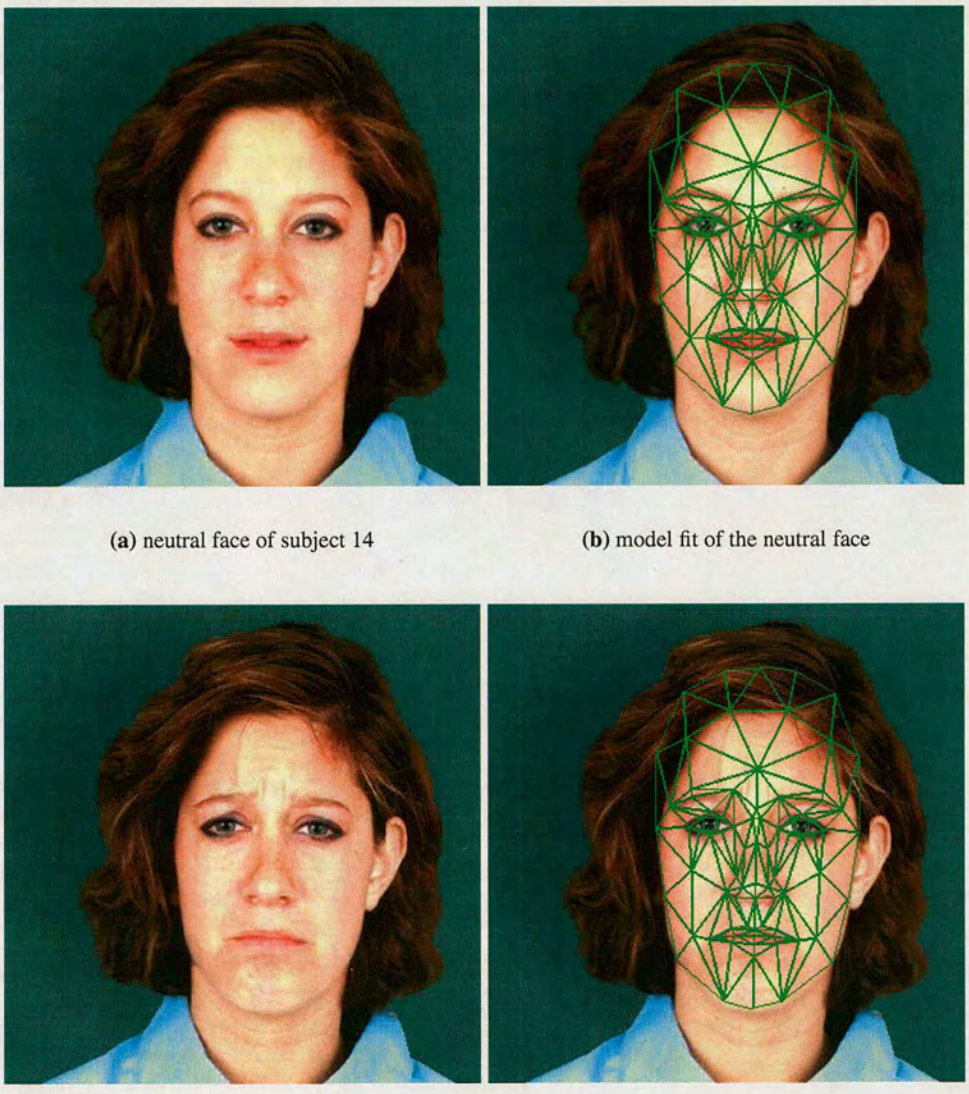
(b) model fit of the neutral face



(c) angry expression of subject 08

(d) model fit of angry expression

Figure 6.31: *Model fit of the neutral and angry face*



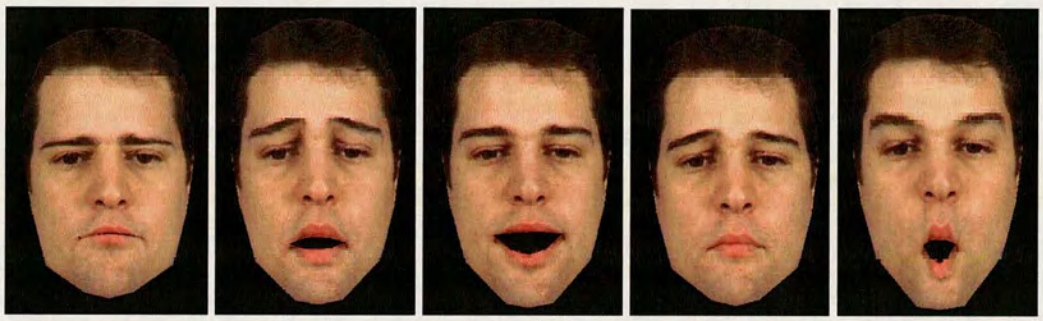
(a) neutral face of subject 14

(b) model fit of the neutral face

(c) sad expression of subject 14

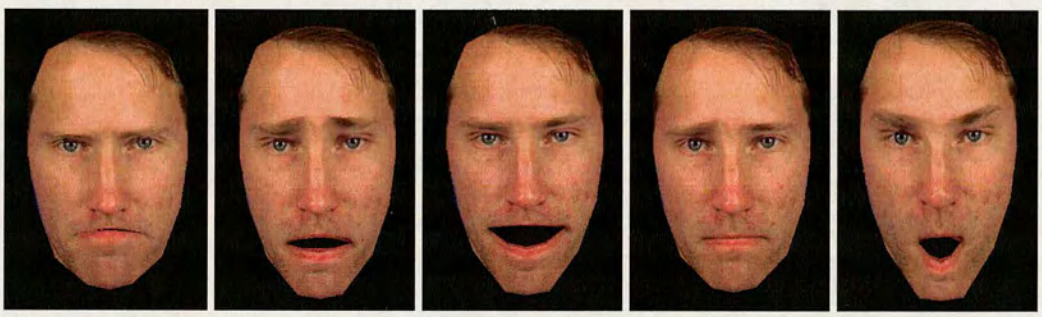
(d) model fit of sad expression

Figure 6.32: *Model fit of the neutral and sad face*



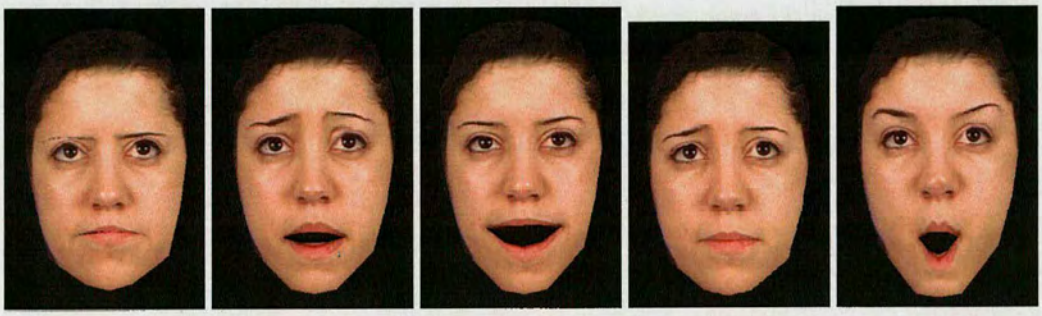
(a) duplicate anger **(b)** duplicate fear **(c)** duplicate joy **(d)** duplicate sadness **(e)** duplicate surprise

Figure 6.33: *Duplication of five major expressions on subject 00021*



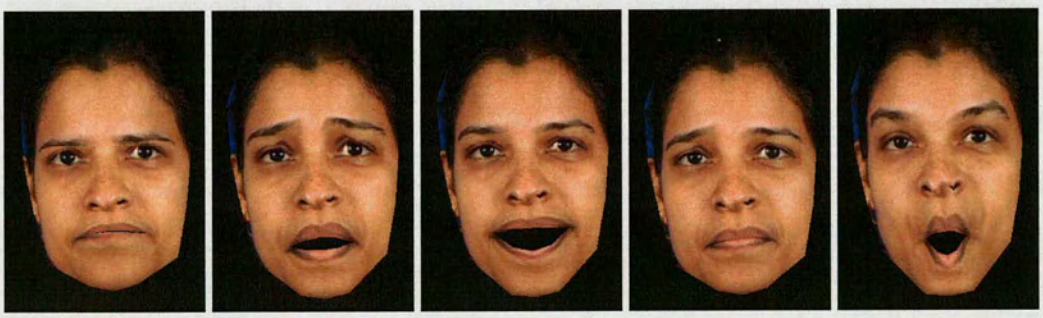
(a) duplicate anger **(b)** duplicate fear **(c)** duplicate joy **(d)** duplicate sadness **(e)** duplicate surprise

Figure 6.34: *Duplication of five major expressions on subject 00111*



(a) duplicate anger **(b)** duplicate fear **(c)** duplicate joy **(d)** duplicate sadness **(e)** duplicate surprise

Figure 6.35: *Duplication of five major expressions on subject 00521*



(a) duplicate anger **(b)** duplicate fear **(c)** duplicate joy **(d)** duplicate sadness **(e)** duplicate surprise

Figure 6.36: *Duplication of five major expressions on subject 02311*

at the feature points which are thus not suitable for animation. This combined appearance and feature measure approach has also been tested by various animations and proved successful.

Although the "Vertex Position Correction" scheme is more accurate, there are several advantages for adopting the second method. First, because the shape/animation units can be extracted after adaptation, these can be transferred to any other different face models as long as they comply with the same shape/animation parameters. Second, since the parameters are transferable, the model can be animated with a specific expression from a particular person, rather than by a set of generic rules. In addition, the extracted shape/animation parameters are valuable for studying facial traits (to recognise age, gender, ethnicity) and expressions (to recognise emotion and intention).

Future work would require depth information for the face. This requires at least two face images from different view points so that the z-coordinates of facial features can be estimated. This would dramatically increase the possible rotation angles of the animated faces. This also helps with the appearance fitting and texture rendering as a single image does not cover the full surface of the face.

Chapter 7
Conclusions

7.1 Achievements of the Thesis

This thesis presents novel and improved algorithms for facial feature fitting and model face fitting. The results of the fitting have been tested with the XM2VTSDB image database [16] and other images and sequences. The fitting of the face model (Candide-3 [17]) was not only assessed by subjective inspection of the wireframe fit of the face, but also by animation using model-based technology. All of the developed fitting algorithms in this thesis are automatic, and considered robust and accurate.

Four important facial features - lip, eyes, eyebrows and chin - are of interest and their contours were fitted. The lips were fitted by an adaptive colour Active Contour Model (Snake) [6]. The algorithm is capable of sampling the lip and skin pixels (in HSI (Hue-Saturation-Intensity) space), thus a colour profile can be built for each individual. The snake incorporates Cohen's balloon model [32] and is able to self-adjust its pressure parameters according to the colour profile. Thus the algorithm is robust against age, gender, ethnicity and lighting conditions. To increase the accuracy of the fitting of the lip corner (as the snake is less competent for getting into sharp corners), a separate lip corner finding algorithm was developed.

Eyes were fitted by an improved Deformable Template algorithm. The iris was fitted first followed by fitting of eye corners and eyelids. A circular template was used for iris fitting as Yuille did [7], but an additional size term, which was derived from the feature geometry, was added in the cost function to avoid template over-shrinking. The eye corners were found by a novel rotating vector approach. The vector, pivoted at the centre of the iris, was rotated to such a point where the area enclosed by its arrowhead contained most white sclera and the eye corner should be in the direction that the vector points. Thus the actual eye corner can be located by a corner detection along this direction. The (upper and lower) eyelids were fitted by using two parabolic templates. The fitting of eyelids is much faster than Yuille's approach [7] as only the templates passing the found eye corners (or their neighbouring points) need to be considered for eyelid fitting. The templates were updated sequentially, which means a smaller

number of parameters are computed simultaneously. These reduce the complexity and number of the iterations thus the speed of fitting is increased dramatically.

Eyebrows were fitted by a cascaded approach consisting of lighting balancing, k-means clustering, inner corner shadow removal and twin snake eyebrow extraction. Because of the curvature of the forehead, the illumination is often unbalanced from left to right in a sub-image containing an eyebrow. Thus linear scaling and offsetting of intensity horizontally was used to re-balance the illumination. This led to more accurate k-means clustering which can divide the image into three regions - skin, dense eyebrow and sparse eyebrow. However, the inner corner of the eyebrow (the area near the nose ridge and eye socket) often contains heavy shadow and cannot be eliminated by the lighting balancing. An edge detection was used to segment the shadow and eyebrow. The twin snake was then deployed in this processed image. The snakes were initialised at the upper and lower image boundaries respectively and moved oppositely to extract the upper and lower boundaries of the eyebrow. The intercept points of the snakes were the eyebrow corners.

The chin was fitted by a novel topologically adaptive snake. This snake is capable of "probing" the area of the image where the snake is going to develop. Two statistical measures of this area, mean and standard deviation of intensities, were computed. The parameters of the snake were self-adjusted according to these measures, therefore the chin can be extracted from various types of skin. In cases where the chin is corrupted by heavy shadow, a novel chin extrapolation technique was developed. The part of the chin which is in shadow is identified by brightness comparison. The initialisation of the snake for this shadowed part of the chin was generated from the linear extrapolation of the unshadowed part. This new initialisation is very close to the true chin outline and thus the chin is fitted with minimal shadow effect.

The face was fitted by a Candide-3 wire-frame model using a combined appearance- and feature- based approach. This novel approach fitted the face by minimising both the appearance and feature errors. It alleviates the drawback of the previous appearance-based approach that is unable to fit facial feature points (on the face model) accurately.

Face animation was generated by applying model-based technology so that the model parameters can be altered. Two sets of animation were generated. The first set followed the MPEG-4 animation rules [143] to generate the five major expressions - anger, fear, joy, sadness and surprise. The second set of animation demonstrated the concept of cloning of expressions. A face

image containing a definite expression [157] was model fitted and thus the parameters related to the expression were extracted. These parameters were then implemented on another model fitted face to produce a similar expression, demonstrating the cloning of the expression from one face to the other. These animations were produced for assessing the model fitting of the face. The results of the animations showed acceptable realism and duplication, indicating a good fit of the face model has been achieved.

7.2 Limitations of the Work

Although the new/improved approaches described in this thesis outperform the previous approaches in many aspects, there are still some limitations existing in the algorithms.

The most noticeable limitation is the speed (time) of the fitting algorithm. Because the facial features are fitted by extracting their entire contours rather than only estimating their locations (as points), the processing time is hugely increased. Generally speaking, each feature takes about $3 \sim 6$ seconds to fit. Thus it will take about 30 seconds to fit all the features in a face. The model fitting of the face takes a longer time to complete, typically 3 minutes are required. If the preprocessing step of face detection and feature localisation is included, the overall processing time for a face image will be about 5 minutes (this is done with a 2.40 GHz Intel(R) Xeon(TM) Processor under Linux system). It is certainly unacceptable for any real-time applications. However, many ways to improve the speed are possible. Firstly, the code (written in C/C++) was not written in an optimised fashion, as the algorithm was coded initially for the purpose of demonstrating the concept of new/improved approaches. It should be possible to rewrite the code in a more efficient way.

Secondly, the algorithm can be modified to improve speed. For example, initialisation is important for active contour models, deformable templates and appearance-based model fitting of the face. An initialisation of the snake closer to the feature can reduce the number of iterations dramatically. A better initial search boundary helps deformable templates to fit the features quicker. A more accurate initial estimation of the face pose accelerates the face model fitting. The computation can become faster by reducing the number of the control points of the snake as well but this may deteriorate the fitting accuracy.

Thirdly, the processing time is related to the size of image. The XM2VTSDB images [16] are very high quality face images (size 720×576), but in real-time applications such as videophone,

the number of pixels in the image is much less (e.g., QCIF, 180×144). The smaller size of the images means less control points of the snake will be used, less pixel distance from the initial to final position, a smaller search space, and less pixels to be compared in PCA.

Finally, better hardware is also helpful. Ahlberg [26] used specialised hardware and showed appearance-based model face tracking can be achieved in real-time. This indicates the processing time of the proposed algorithms in this thesis can be cut down significantly if specialised hardware is used.

Other limitations include simplicity of the face model (Candide-3) and the proposed algorithms are unable to handle large head rotations. The simplicity of Candide-3 is convenient for demonstrating the concepts of face model fitting and model-based animation, but this simplicity also impedes producing animations that are commercially acceptable. It is because some facial features are missing as well as the number of vertices and polygons of the model being small. For example, Candide-3 has no teeth, tongue, ear and hair. No teeth and tongue means that animation of speech cannot be realistic. No ear and hair means that it is impossible to attach personal accessories, such as earrings or hairpins to the model. The low number of surfaces results in a jagged face boundary and affects the smoothness of animation. However, this can be solved by adopting one of the high complexity models used commercially. This will increase the number of feature points extracted for model face fitting, but it should not be a problem as the contours of the features are known.

Most of the techniques in this work are based on the assumption that the face is nearly-frontal. It is important because most of the initialisation of the algorithms is set up using the geometric relationships between the features. Such relationships become invalid if the rotation of the face is large. To improve this, estimation of the face pose (rotation about x- y- and z- axes) in the early stage is necessary. Thus the geometric relationships between the features can be calculated by rotating the face back to a frontal view.

Furthermore, some limitations are also identified in each fitting technique. For lip fitting, facial hair and revealed teeth are two major factors causing the system to fail. Dense beards and moustaches (but not stubble) create strong intensity edges that overpower the snake force generated by the lip-skin colour difference and they may also occlude the true lip edge. These cause the snake to fit the beard/moustache contour rather than the lip contour. Revealed teeth significantly alter the expected colour profile as the algorithm used only expects the presence

of skin and lip colours in the search region. This results in a wrong colour threshold estimation for lips and skin, making the snake unable to fit the lip contour.

For eye fitting, glasses and reflections are two major factors causing the system to fail. The glasses introduce frame edges which are stronger than the edges of the eye features. Reflections can generate unexpected high intensity, low saturation areas in the eye search region. Reflection problems are also intensified by glasses as they often have higher reflectivity. As the cost functions for the deformable templates use edge, intensity and saturation terms, these frame edges and reflections can interfere with the template deformation, resulting in a misfit of the eye features.

For eyebrow fitting, glasses, hair and unbalanced lighting are the major factors causing the system to fail. The glasses and hair introduce additional edges as well as occluding the eyebrow, making the snakes fit the edges of the hair or the glasses frame instead of the eyebrow contour. Furthermore, although a linear lighting balancing mechanism is used prior to the eyebrow fitting, this is not always sufficient to compensate for a wide range of illumination. Also, the shadow is more complex in largely side-lit images, such cases can not be approximated by a linear model.

Chin fitting has the limitation of facial hair occlusion. Because dense beards can cover the chin characteristic found in Section 5.2, the snake is unable to fit the chin. The technique also suffers from the general limitation of being unable to handle large head rotation. This is because the search region is defined by facial feature geometry and would not be able to fully encompass the chin if the head rotation is too large.

Other limitations of the model face fitting technique include no depth information and some missing face texture. These are inevitable as only a single face image is used. In consequence, the face cannot be rotated to larger angles or the artefacts (e.g. unrealistic nose height and some void surface patches) will be visible in an animation.

7.3 Future Research

Further improvements on facial feature fitting and face model fitting are also possible. As lip fitting is frequently degraded by facial hair and revealed teeth, detection and segmentation of facial hair and teeth are suggested. Facial hair may be identified by using texture properties and

thus the coverage of the hair may be estimated. Then a colour model of the facial hair could be established and possibly incorporated into the colour snake in conjunction with the skin and lip colour models. In consequence, the snake would be able to recognise the lip-skin boundary using the lip and skin colour models (if the lip is not occluded by the facial hair) and the lip-facial hair boundary using the lip and facial hair colour models (if the lip is partially occluded by the facial hair).

The teeth can be discriminated from the skin and lip by examining the saturation profile, as the teeth are whiter which means lower saturation. Identification of the teeth should result in a better estimation of the skin-lip colour thresholds (as the teeth pixels are removed from the threshold calculation) thus the lip outline can be fitted more accurately. This also indicates a possibility of finding the inner boundary of the lip if the teeth could be detected and segmented.

Eye fitting is frequently interfered with by glasses frames and reflections. To reduce such interferences, a reflection cancellation algorithm and a method to remove the glasses frames are suggested. The glasses frames may be removed by using traditional edge detection as the frames generally produce very distinct edges. Reflections in the eye region may be identified by using Perez et al's [114] technique to detect abnormally high (or change of) saturation.

Hair, glasses and unbalanced lighting result in unsatisfactory eyebrow fitting. Again, the glasses frames may be identified by tradition edge detection, while the hair may be removed by using texture properties. Fitting the eyebrow in the heavy shadow resulted from largely side-lit images is challenging. A possible solution may be to detect the direction of the light sources and use the topology of the eyebrow surface [158] to apply a more sophisticated (non-linear) lighting balancing scheme. Understanding eyebrow texture may be also beneficial in fitting the shadowed eyebrow region.

The major limitation of the chin fitting is the occlusion of facial hair. Depending upon the extent of the facial hair coverage, the chin may only be fitted when there is a sufficient area of the chin visible (chins not perceptible by human eyes cannot possibly be fitted). Partially occluded chins may be fitted by recognising the area of the facial hair (this may require texture recognition). Then the visible parts of the chin may be fitted followed by an estimation of the invisible parts.

Limitations of the face model fitting such as no depth information and missing face texture have been noted. To solve these, a set of face images taken at different view angles may be used. At

least two images are required to produce a stereo view of an image object. The estimation of the z-coordinate may be further improved if more cameras are used [159]. Multiple images should also result in more complete face textures. A texture blending technique similar to Pighin et. al [138] is suggested to produce more realistic and smooth face texture. Their technique blended the textures from different images by assigning different weights according to the angle between the face surface normal and the camera direction. In addition, if a 3D scanner is available, better depth information and full face texture can be obtained.

The results of this work open up many future research opportunities. The animation units extracted from the face model can be used for expression recognition and classification [2]. This leads to more psychological and medical uses. Examples are micro- and subtle- expression recognition, lie detection, emotion and intention analysis, cognitive processes, personality studies and early child development [3]. The shape units extracted from the face model can be used for personal identity recognition, such as recognising age, gender, ethnic origin and condition of health [1].

Other future research related to this work includes lip-reading (lip fitting) [30, 89, 90] and gaze detection (eye fitting) [106, 118–121]. Both studies have potential applications in the field of HCI (Human Computer Interface). Lip-reading requires finding both the outer and inner lip boundaries and performing temporal lip tracking. Such a system is expected to handle the revealed teeth and tongue so the inner lip boundary can be extracted. The system also requires matching the tracked lip to a sequence of phonemes so that speech recognition can be performed.

Gaze detection and tracking are popular because it can be used for hands-free computer display control and interaction. However, as direction of gaze changes in relation to the small movement of the eyeball, the accuracy of estimated direction of gaze from the images is still unsatisfactory. Another issue for gaze tracking is blinking handling, as the tracking may be lost because irises/ pupils are invisible when the blinking occurs. These two concerns are the major bottlenecks to prevent these techniques being used in commercial applications. Another valuable application for using gaze and blinking is for drowsiness detection.

7.4 Final Remarks

Facial feature fitting and Model face fitting are important research areas with a wide range of applications across various disciplines. As a result, the algorithms developed in this work are valuable and can be used as a basis for future research. The suggested improvements, when combined with the algorithms' existing attributes of automation, robustness and accuracy, make a fully usable face modelling system foreseeable. This will directly benefit use of face modelling in the IT, psychology and medicine sectors and also benefit applications using individual facial features such as lips (lip-reading) and eyes (gaze control, drowsiness detection).

Appendix A
Publications

P.P. Kuo, P. Hillman and J. M. Hannah. "Improved Lip Fitting and Tracking for Model-based Multimedia and Coding", in Proceedings of Visual Information Engineering (VIE2005), IEE, Glasgow, pp251-258, Apr. 2005.

P. P. Kuo and J. M. Hannah. "An Improved Eye Feature Extraction Algorithm Based on Deformable Templates", in Proceedings of International Conference on Image Processing (ICIP2005), IEEE, Genova, Vol. II, pp1206-1209, Sep. 2005.

P. P. Kuo, P. Hillman and J. M. Hannah. "Improved Facial Feature Extraction for Model-Based Multimedia", in Proceedings of the European Conference on Visual Media Production (CVMP2005), IEE, London, pp137-146, Nov. 2005.

P. P. Kuo and J. M. Hannah. "Improved Chin Fitting Algorithm Based on An Adaptive Snake", to appear in Proceedings of International Conference on Image Processing (ICIP2006), IEEE, Atlanta, Oct. 2006.

P. Hillman, P.P. Kuo and J. M. Hannah. "Hybrid Facial Model Fitting Using Active Appearance Models and Contour-Based Facial Feature Location", to appear in Proceedings of International Conference on Image Processing (ICIP2006), IEEE, Atlanta, Oct. 2006.

Appendix B
Candide-3 and MPEG-4 Conversion

Vertex	Description	MPEG-4 FFP	
0	Top of skull	11	4
1	(Middle border between hair and forehead)	11	1
2	Middle of forehead		
3	Midpoint between eyebrows		
4	Not used (replaced by 77 and 78 in CANDIDE-1)		
5	Nose tip	9	3
6	Bottom middle edge of nose	9	15
7	Middle point of outer upper lip contour	8	1
8	Middle point of outer lower lip contour	8	2
9	Chin boss	2	10
10	Bottom of the chin	2	1
11	Left of top of skull		
12	Left of top of skull		
13	(Left border between hair and forehead)	11	3
14	Left side of skull		
15	Outer corner of left eyebrow	4	5
16	Uppermost point of the left eyebrow	4	3
17	Inner corner of left eyebrow	4	1
18	Lower contour of the left eyebrow, straight under 16		
19	Center of upper outer left eyelid	3	13
20	Outer corner of left eye	3	7
21	Center of upper inner left eyelid	3	1
22	Center of lower inner left eyelid	3	3
23	Inner corner of left eye	3	11
24	Center of lower outer left eyelid	3	9
25	Left nose border		
26	Left nostril outer border	9	1
27	Left cheek bone	5	3
28	Inner contact point between left ear and face	10	8
29	Upper contact point between left ear and face	10	9
30	Left corner of jaw bone	2	13
31	Left corner of outer lip contour	8	3
32	Chin left corner	2	11
33	Uppermost point of left outer lip contour	8	10
34	(Middle border between hair and forehead)	11	1
35	Not used (identical to 2)		
36	Not used (identical to 3)		
37	Not used (identical to 4)		
38	Not used (identical to 5)		

Vertex	Description	MPEG-4 FFP	
39	Not used (identical to 6)		
40	Middle point of inner lower lip contour	2	3
41	Not used (identical to 8)		
42	Not used (identical to 9)		
43	Not used (identical to 10)		
44	Right of top of skull		
45	Right of top of skull		
46	(Right border between hair and forehead)	11	2
47	Right side of skull		
48	Outer corner of right eyebrow	4	6
49	Uppermost point of the right eyebrow	4	4
50	Inner corner of right eyebrow	4	2
51	Lower contour of the right eyebrow, straight under 49		
52	Center of upper outer right eyelid	3	14
53	Outer corner of right eye	3	12
54	Center of upper inner right eyelid	3	2
55	Center of lower inner right eyelid	3	4
56	Inner corner of right eye	3	8
57	Center of lower outer right eyelid	3	10
58	Right nose border		
59	Right nostril border	9	2
60	Right cheek bone	5	4
61	Lower contact point between right ear and face	10	7
62	Upper contact point between right ear and face	10	10
63	Right corner of jaw bone	2	14
64	Right corner of outer lip contour	8	4
65	Chin right corner	2	12
66	Uppermost point of right outer lip contour	8	9
67	Left iris, outer upper corner of bounding (square) box		
68	Left iris, outer lower corner of bounding (square) box		
69	Right iris, outer upper corner of bounding (square) box		
70	Right iris, outer lower corner of bounding (square) box		
71	Left iris, inner upper corner of bounding (square) box		
72	Left iris, inner lower corner of bounding (square) box		
73	Right iris, inner upper corner of bounding (square) box		
74	Right iris, inner lower corner of bounding (square) box		
75	Left side of nose tip		
76	Right side of nose tip		
77	Left upper edge of nose bone	9	7
78	Right upper edge of nose bone	9	6
79	Midpoint between FFP 8.4 and 8.1 on outer upper lip contour	8	5

Vertex	Description	MPEG-4 FFP	
80	Midpoint between FFP 8.3 and 8.1 on outer upper lip contour	8	6
81	Midpoint between FFP 2.2 and 2.5 on the inner upper lip contour	2	6
82	Midpoint between FFP 2.2 and 2.4 on the inner upper lip contour	2	7
83	Midpoint between FFP 2.3 and 2.5 on the inner lower lip contour	2	8
84	Midpoint between FFP 2.3 and 2.4 on the inner lower lip contour	2	9
85	Midpoint between FFP 8.4 and 8.2 on outer lower lip contour	8	7
86	Midpoint between FFP 8.3 and 8.2 on outer lower lip contour	8	8
87	Middle point of inner upper lip contour	2	2
88	Left corner of inner lip contour	2	4
89	Right corner of inner lip contour	2	5
90	Center of the left cheek	5	1
91	Center of the right cheek	5	2
92	Left lower edge of nose bone	9	13
93	Right lower edge of nose bone	9	14
94	Middle lower edge of nose bone (or nose bump)	9	12
95	Outer upper edge of left upper eyelid		
96	Outer upper edge of right upper eyelid		
97	Outer lower edge of left upper eyelid		
98	Outer lower edge of right upper eyelid		
99	Outer upper edge of left lower eyelid		
100	Outer upper edge of right lower eyelid		
101	Outer lower edge of left lower eyelid		
102	Outer lower edge of right lower eyelid		
103	Inner upper edge of left upper eyelid		
104	Inner upper edge of right upper eyelid		
105	Inner lower edge of left upper eyelid		
106	Inner lower edge of right upper eyelid		
107	Inner upper edge of left lower eyelid		
108	Inner upper edge of right lower eyelid		
109	Inner lower edge of left lower eyelid		
110	Inner lower edge of right lower eyelid		
111	Bottom left edge of nose	9	5
112	Bottom right edge of nose	9	4

Appendix C

Candide-3 Shape and Animation Units and the Interpretation

C.1 Shape Units

Shape Unit (AU)	Description
SU 0	Head height
SU 1	Eyebrows vertical position
SU 2	Eyes vertical position
SU 3	Eyes, width
SU 4	Eyes, height
SU 5	Eye separation distance
SU 6	Cheeks z
SU 7	Nose z-extension
SU 8	Nose vertical position
SU 9	Nose, pointing up
SU 10	Mouth vertical position
SU 11	Mouth width
SU 12	Eyes vertical difference
SU 13	Chin width

*Number of Shape Units: 14

C.2 Animation Units

Animation Unit (AU)	Description
AU 0	Upper lip raiser (AUV 0)
AU 1	Jaw drop (AUV11)
AU 2	Lip stretcher (AUV 2)
AU 3	Brow lowerer (AUV 3)
AU 4	Lip corner depressor (AUV14)
AU 5	Outer brow raiser (AUV 5)
AU 6	Eyes closed (AUV 6)
AU 7	Lid tightener (AUV 7)
AU 8	Nose wrinkler (AUV 8)
AU 9	Lip presser (AUV 9)
AU 10	Upper lid raiser (AUV10)
AU 11 (FAP 3)	open_jaw
AU 12 (FAP 4)	lower_t_midlip
AU 13 (FAP 5)	raise_b_midlip
AU 14 (FAP 6)	stretch_l_cornerlip
AU 15 (FAP 7)	stretch_r_cornerlip
AU 16 (FAP 8)	lower_t_lip_lm
AU 17 (FAP 9)	lower_t_lip_rm
AU 18 (FAP10)	raise_b_lip_lm
AU 19 (FAP11)	raise_b_lip_rm
AU 20 (FAP12)	raise_l_cornerlip
AU 21 (FAP13)	raise_r_cornerlip
AU 22 (FAP14)	thrust_jaw
AU 23 (FAP15)	shift_jaw
AU 24 (FAP16)	push_b_lip
AU 25 (FAP17)	push_t_lip

Animation Unit (AU)	Description
AU 26 (FAP18)	depress_chin
AU 27 (FAP19)	close_t_l_eyelid
AU 28 (FAP20)	close_t_r_eyelid
AU 29 (FAP21)	close_b_l_eyelid
AU 30 (FAP22)	close_b_r_eyelid
AU 31 (FAP23)	yaw_l_eyeball
AU 32 (FAP24)	yaw_r_eyeball
AU 33 (FAP25)	pitch_l_eyeball
AU 34 (FAP26)	pitch_r_eyeball
AU 35 (FAP27)	thrust_l_eyeball
AU 36 (FAP28)	thrust_r_eyeball
AU 37 (FAP29)	dilate_l_pupil
AU 38 (FAP30)	dilate_r_pupil
AU 39 (FAP31)	raise_l_i_eyebrow
AU 40 (FAP32)	raise_r_i_eyebrow
AU 41 (FAP33)	raise_l_m_eyebrow
AU 42 (FAP34)	raise_r_m_eyebrow
AU 43 (FAP35)	raise_l_o_eyebrow
AU 44 (FAP36)	raise_r_o_eyebrow
AU 45 (FAP37)	squeeze_l_eyebrow
AU 46 (FAP38)	squeeze_r_eyebrow
AU 47 (FAP39)	puff_l_cheek
AU 48 (FAP40)	puff_r_cheek
AU 49 (FAP41)	lift_l_cheek
AU 50 (FAP42)	lift_r_cheek

Animation Unit (AU)	Description
AU 51 (FAP51)	lower_t_midlip_o
AU 52 (FAP52)	raise_b_midlip_o
AU 53 (FAP53)	stretch_l_cornerlip_o
AU 54 (FAP54)	stretch_r_cornerlip_o
AU 55 (FAP55)	lower_t_lip_lm_o
AU 56 (FAP56)	lower_t_lip_rm_o
AU 57 (FAP57)	raise_b_lip_lm_o
AU 58 (FAP58)	raise_b_lip_rm_o
AU 59 (FAP59)	raise_l_cornerlip_o
AU 60 (FAP60)	raise_r_cornerlip_o
AU 61 (FAP61)	stretch_l_nose
AU 62 (FAP62)	stretch_r_nose
AU 63 (FAP63)	raise_nose
AU 64 (FAP64)	bend_nose

*Number of Animation Units: 65

**l = left, r = right, t = top, b = bottom, i = inner, o = outer, m = middle

References

[1] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Annual Conference Series, Computer Graphics, Siggraph*, pp. 187–194, August 1999.

[2] M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Measuring facial expressions by computer image analysis," *Psychophysiology*, vol. 36, pp. 253–263, 1999.

[3] P. Ekman, ed., *What the face reveals: basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. New York, USA: Oxford University Press, 1997.

[4] P. Hillman, J. M. Hannah, and P. M. Grant, "Global fitting of a facial model to facial features for model-based video coding," in *Proceedings of the 3rd International symposium on Image and Signal Processing and Analysis (ISPA)*, vol. 1, pp. 359–364, 2003.

[5] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes:active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

[7] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 99–111, 1992.

[8] P. Kuo, P. Hillman, and J. M. Hannah, "Improved lip fitting and tracking for model-based multimedia and coding," in *Proceedings of visual information engineering (VIE), IEE*, (Glasgow, UK), pp. 251–258, IEE, April 2005.

[9] P. Kuo and J. M. Hannah, "An improved eye feature extraction algorithm based on deformable templates," in *Proceedings of International Conference on Image Processing (ICIP)*, (Genva, Italy), pp. 1206–1209, IEEE, Sep 2005.

[10] P. Kuo, P. Hillman, and J. M. Hannah, "Improved facial feature extraction for model-based multimedia," in *Proceedings of the European Conference on Visual Media Production, IEE*, (London, UK), pp. 137–146, IEE, Nov 2005.

[11] P. Kuo and J. M. Hannah, "Improved chin fitting algorithm based on an adaptive snake," in *Proceedings of International Conference on Image Processing*, (Atlanta, GA), IEEE, Oct 2006. Accept for Publish.

[12] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[13] J. Ahlberg, *Model-Based Coding: Extraction, Coding and Evaluation of Face Model Parameters*. PhD thesis, University of Linköping, Linköping, Sweden, 2002.

[14] P. Hillman, P. Kuo, and J. M. Hannah, "Hybrid facial model fitting using active appearance models and contour-based facial feature location," in *Proceedings of International Conference on Image Processing*, (Atlanta, GA), IEEE, Oct 2006. Accept for Publish.

[15] I. T. Jolliffe, *Principal Component Analysis*. Springer Series in Statistics, 2 ed., 2002.

[16] K. Messer, J. Matas, J. Kittler, and K. Jonsson, "Xm2vtsdb: The extended m2vts database." in Audio- and Video-based Biometric Person Authentication, AVBPA1999, 1999.

[17] J. Ahlberg, "Candide-3 - an updated parametrised face," Tech. Rep. LiTH-ISY-R-2326, Image Coding Group, Dept. of Electrical Engineering, Linköping University, Sweden, 2001.

[18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison Wesley, Sept 1993.

[19] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*. Oxford: Newnes, 2002.

[20] http://www.netnam.vn/unescocourse/computervision/12.htm.

[21] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 586–591, IEEE, 1991.

[22] http://www.owlnet.rice.edu/ elec301/Projects99/faces/images.html.

[23] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[24] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for medical image analysis and computer vision," *Proceeding of SPIE Medical Imaging*, 2001.

[25] T. F. Cootets and C. J. Taylor, "Constrained active appearance models," in *Proceeding of International Conference on Computer Vision*, (Vancouver), pp. 748–754, IEEE, July 2001.

[26] J. Ahlberg, "An active model for facial feature tracking," *EURASIP J. Appl. Signal Processing*, vol. 6, pp. 566–571, 2002.

[27] K. H. Seo, W. Kim, C. Oh, and J. J. Lee, "Face detection and facial feature extraction using color snake," in *Proceedings of the IEEE International Symposium*, vol. 2, pp. 457–462, 2002.

[28] D. Sun and L. Wu, "Face boundary extraction by statistical constraint active contour model," in *Proceedings of International Conference on Neural Networks and Signal Processing 2003*, vol. 2, pp. 14–17, IEEE, Dec 2003.

[29] F. Hara and K. Tanaka, "Automatic feature extraction of facial organs and contour," in *Proceedings of International Workshop on Robot and Human Communication*, pp. 386–390, IEEE, 1997.

[30] B. Mark, E. J. Holden, and O. Robyn, "Lip tracking using pattern matching snakes," in *Proceedings of The 5th Asian Conference on Computer Vision ACCV*, pp. 273–278, 2002.

[31] M. Okubo and T. Watanabe, "Lip motion capture and its application to 3-d molding," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 187–192, 1998.

[32] L. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991.

[33] L. D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-d and 3-d images," *Translations on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 11331–1147, 1993.

[34] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snake," in *Proceedings of Computer Vision and Pattern Recognition (CVPR97)*, pp. 66–71, IEEE, 1997.

[35] K. H. Seo and J. J. Lee, "Object tracking using adaptive color snake model," in *Proceedings of the IEEE/ASME International Conference*, vol. 2, pp. 1406–1410, 2003.

[36] H. Schaub and C. E. Smith, "Color snakes for dynamic lighting conditions on mobile manipulation platforms," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, (Las Vegas, NV), pp. 1272–1277, 2003.

[37] M. Kampmann, "Automatic 3-d face model adaptation for model-based coding of videophone sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 3, pp. 172–182, 2002.

[38] Z. Wu, P. S. Aleksic, and A. K. Katsaggelos, "Lip tracking for mpeg-4 facial animation," in *Proceedings of International Conference on Multimodal Interfaces (ICMI)*, (Pittsburgh, PA), IEEE, Oct 2002.

[39] L. Zhang, "Estimation of eye and mouth corner point positions in a knowledge-based coding system," *Digital Compression Technologies and Systems for Video Communications*, vol. 2952, pp. 21–28, Oct 1996.

[40] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, Jan 2002.

[41] G. Yang and T. S. Huang, "Human face detection in a complex background," *Pattern Recognition*, vol. 27, no. 1, pp. 53–63, 1994.

[42] C. Kotropoulos and A. T. and, "Rule-based face detection in frontal view," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 2537–2540, IEEE, 1997.

[43] T. Kanade, *Picture Processing by Computer Complex and Recognition of Human faces*. PhD thesis, Kyoto University, 1973.

[44] T. K. Leung, M. C. Burl, and P. Perona, "Finding faces in cluttered scenes using random labelled graph matching," in *Proceedings of International Conference on Computer Vision*, pp. 637–644, IEEE, 1995.

[45] K. C. Yow and R. Cipolla, "Feature-based human face detection," *Journal of Image and Vision Computing*, vol. 15, no. 9, pp. 713–735, 1997.

[46] Y. Dai and Y. Nakano, "Face-texture model-based on sgld and its application in face detection in a color scene," *Journal of Pattern Recognition*, vol. 29, no. 6, pp. 1007–1017, 1996.

[47] J. Yang and A. Waibel, "A real-time face tracker," in *Proceedings of Workshop on Applications of Computer Vision*, pp. 142–147, IEEE, 1996.

[48] S. Mckenna and S. G. an Y. Raja, "Modelling facial colour and identity with gaussian mixtures," *Journal of Pattern Recognition*, vol. 31, no. 12, pp. 1883–1892, 1998.

[49] R. Kjeldsen and J. Kender, "Finding skin in color images," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 312–317, IEEE, 1996.

[50] K. C. Yow and R. Cipolla, "A probabilistic framework for perceptual grouping of features for human face detection," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 16–21, IEEE, 1996.

[51] K. C. Yow and R. Cipolla, "Enhancing human face detection using motion and active contour," in *Proceeding of Asian Conference on Computer Vision*, pp. 515–522, IEEE, 1998.

[52] M. F. Augusteijn and T. L. Skujca, "Identification of human faces through texture-based feature recognition and neural network technology," in *Proceedings of Conference on Neural Networks*, pp. 392–398, IEEE, 1993.

[53] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification," *IEEE Transactions on Systems, Man, Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[54] T. Kohonen, *Self-Organization and Associative Memory*. Springer, 1989.

[55] H. P. Graf, T. Chen, E. Petajan, and E. Cosatto, "Locating faces and facial parts," in *Proceedings of International Workshop on Automatic Face and Gesture recognition*, pp. 41–46, IEEE, 1995.

[56] H. P. Graf, E. C. D. Gibbon, M. Kocheisen, and E. Petajan, "Multimodel system for locating heads and faces," in *Proceedings of International Workshop on Automatic Face and Gesture recognition*, pp. 88–93, IEEE, 1996.

[57] T. S. Jebara and A. Pentland, "Parametrized structure from motion for 3d adaptive feedback tracking of faces," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 144–150, IEEE, 1997.

[58] T. S. Jebara, K. Russell, and A. Pentland, "Mixtures of eigenfeatures for real-time structure form texture," in *Proceedings of International Conference on Computer Vision*, pp. 128–135, IEEE, 1998.

[59] Y. Miyake, H. Saitoh, H. Yaguchi, and N. Tsukada, "Facial pattern detection and color correction from television picture for newspaper printing," *Journal of Imaging technology*, vol. 16, no. 5, pp. 165–169, 1990.

[60] J. L. Crowley and J. M. Bedrune, "Integration and control of reactive visual processes," in *Proceedings of European Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 47–58, 1994.

[61] N. Oliver, A. Pentland, and F. Berard, "Lafer: Lips an face real time tracker," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 123–129, IEEE, 1997.

[62] D. Saxe and R. Foulds, "Toward robust skin identification in video images," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 379–384, IEEE, 1996.

[63] K. Sobottka and I. Pitas, "Face localization and feature extraction based on shape and color information," in *Proceedings of International Conference on Image Processing*, pp. 483–486, IEEE, 1996.

[64] K. Sobottka and I. Pitas, "Segmentation and tracking of faces in color images," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 236–241, IEEE, 1996.

[65] H. Wang and S. F. Chang, "A highly efficient system for automatic face region detection in mpeg video," *Journal of IEEE Transactions on circuits and Systems for Video Technology*, vol. 7, no. 4, pp. 615–618, 1997.

[66] D. Chai and K. N. Ngan, "Locating facial region of a head-and-shoulders color image," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 124–129, IEEE, 1998.

[67] Q. Chen, H. Wu, and M. Yachida, "Face detection by fuzzy matching," in *Proceedings of International Conference on Computer Vision*, pp. 591–596, IEEE, 1995.

[68] M. H. Yang and N. Ahuja, "Detecting human faces in color images," in *Proceedings of International Conference on Image Processing*, vol. 1, pp. 127–130, IEEE, 1998.

[69] E. Saber and A. M. Tekalp, "Frontal-view face detection and facial feature extraction using color, shape, and symmetry based cost function," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 669–680, 1998.

[70] T. Sakai, M. Nagao, and S. Fujibayashi, "Line extraction and pattern detection in a photograph," *Pattern Recognition*, vol. 1, pp. 233–248, 1969.

[71] I. Craw, H. Ellis, and J. Lishman, "Automatic extraction of face features," *Pattern Recognition Letters*, vol. 5, pp. 183–187, 1987.

[72] I. Craw, D. Tock, and A. Bennett, "Finding face features," in *Proceedings of European Conference on Computer Vision*, pp. 92–96, 1992.

[73] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," in *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition*, (San Diego, CA, USA), pp. 104–109, IEEE, June 1989.

[74] A. Yuille, "Deformable templates for face recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 59–70, 1991.

[75] K. Lam and H. Yan, "Fast algorithm for locating head boundaries," *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 351–359, 1994.

[76] Y. H. Kwon and N. D. V. Lobo, "Face detection using templates," in *Proceedings of International Conference on Pattern Recognition*, pp. 764–767, IEEE, 1994.

[77] T. F. Cootes and C. J. Taylor, "Locating faces using statistical feature detectors," in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 204–209, IEEE, 1996.

[78] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Learning to identity and track faces in image sequences," in *Proceedings of International Conference on Computer Vision*, pp. 317–322, IEEE, 1998.

[79] A. Lanitis, C. J. Taylor, and T. F. Cootes, "An automatic face identification system using flexible appearance models," *Journal of Image and Vision Computing*, vol. 13, no. 5, pp. 393–401, 1995.

[80] T. Agui, Y. Kokubo, H. Nagashi, and T. Nagao, "Extraction of face recognition from monochromatic photographs using neural networks," in *Proceedings of International Conference on Automation, Robotics and Computer Vision*, vol. 1, pp. 18.8.1–18.8.5, 1992.

[81] M. Propp and A. Samal, "Artificial neural network architectures for human face detection," *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 2, pp. 535–540, Nov 1992.

[82] R. V. C. Monrocq and Y. L. Cun, "An original approach for the localisation of objects in images," *Proceedings of Vision, Image and Signal Processing*, vol. 141, pp. 245–250, 1994.

[83] G. Burel and D. Carel, "Detection and localization of faces on digital images," *Pattern Recondition Letters*, vol. 15, no. 10, pp. 963–967, 1994.

[84] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proceedings of Conference on Compute Vision and Pattern Recognition*, pp. 130–136, IEEE, 1997.

[85] Y. Yokoo and M. Hagiwara, "Human faces detection method using genetic algorithm," in *Proceedings of International Conferences on Evolutionary Computation*, pp. 113–118, IEEE, May 1996.

[86] H. Schneiderman and T. Kanade, "Probabilistic modelling of local appearance and spatial relationships for object recognition," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 45–51, IEEE, 1998.

[87] A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri, "Finding faces in photographs," in *Proceedings on International Conference on Computer Vision*, pp. 640–645, IEEE, 1998.

[88] L. Zhang, "Estimation of the mouth features using deformable templates," *Proceedings of International Conference on Image Processing, Santa Barbara, USA*, pp. 26–29, 1997.

[89] K. Sugahara, M. Kishino, and R. Konishi, "Personal computer based real time lip reading system," in *Proceedings of International Conference on Signal Processing Proceedings (WCCC-ICSP)*, vol. 2, pp. 1341–1346, 2000.

[90] T. Shinchi, Y. Maeda, K. Sugahara, and R. Konishi, "Vowel recognition according to lip shapes by using neural network," in *Proceedings of the IEEE International Joint Conference on Neural Networks Proceedings and IEEE World Congress on Computational Intelligence.*, vol. 3, pp. 1772–1777, 1998.

[91] M. J. T. Reinders, F. A. Odijk, J. van der Lubbe, and J.J.Gerbrands, "Tracking of global motion and facial expressions of a human face in image sequences," in *Proceedings of the Conference on Visual Communications and Image Processing*, (Cambridge MA), pp. 1516–1527, 1993.

[92] R. L. Rudianto and K. N. Ngan, "Automatic 3d wireframe model fitting to frontal facial image in model-based video coding," in *Proceedings of Picture Coding Symposium (PCS)*, (Melbourne, Australia), pp. 585–588, 1996.

[93] P. Delmas, N. Eveno, and M. Lievin, "Towards robust lip tracking," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 528–531, 2002.

[94] P. Delmas, P. Coulon, and V. Fristot, "Automatic snakes for robust lip boundaries extraction," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 6, pp. 3069–3072, 1999.

[95] N. Eveno, A. Caplier, and P. Coulon, "Jumping snakes and parametric model for lip segmentation," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 867–870, 2003.

[96] A. M. Peacock, *Information Fusion for Improved Motion Estimation*. PhD thesis, University of Edinburgh, Edinburgh, May 2001.

[97] P. M. Antoszczyszyn, J. M. Hannah, and P. M. Grant, "Accurate automatic frame fitting for semantic-based moving image coding using a facial code-book," in *Proceedings of International Conference on Image Processing*, vol. 1, pp. 689–692, IEEE, 1996.

[98] P. C. R. Lanzarotti, "Fiducial point localization in color images of face foregrounds," *Journal of Image and vision Computing*, no. 22, pp. 863–872, 2004.

[99] S. K. nad J. Ohya, "Two-step approach for real-time eye tracking with a new filtering technique," in *Proceedings of International Conference on Systems, Man and Cybernetics (ICSMC2000)*, vol. II, pp. 1366–1371, IEEE, Oct 2000.

[100] D. W. Hansen and A. E. C. Pece, "Iris tracking with feature contours," in *Proceedings of International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 208–214, IEEE, Oct 2003.

[101] S. Ramadan, W. Abd-almageed, and C. E. Smith, "Eye tracking using active deformable models," in *Proceedings on the III Indian Conference on Computer Vision, Graphics and Image Processing*, (India), Dec 2002.

[102] Y. L. Tian, T. Kanade, and J. Cohn, "Dual-state parametric eye tracking," in *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 110–115, IEEE, 2000.

[103] Z. Hammal and A. Caplier, "Eyes and eyebrows parametric models for automatic segmentation," in *Proceedings of 6th IEEE South-west Symposium on Image Analysis and Interpretation*, pp. 138–141, IEEE, March 2004.

[104] C. H. Su, Y. S. Chen, Y. P. Hung, C. S. Chen, and J. H. Chen, "A real-time robust eye tracking system for autostereoscopic displays using stereo cameras," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Taipei, Taiwan), pp. 1677–1681, IEEE, Sep 2003.

[105] S. Amarnag, R. S. Kumaran, and J. N. Gowdy, "Real time eye tracking for human computer interfaces," in *Proceedings of International Conference on Multimedia and Expo. (ICME2003)*, vol. III, (Baltimore), pp. 557–560, IEEE, July 2003.

[106] A. Haro, M. Flickner, and I. Essa, "Detecting and tracking eyes by using their physiological properties, dynamics, and appearance," in *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 163–168, IEEE, June 2000.

[107] J. Daugman, "How iris recognition works," in *Proceedings of International Conference on Image Processing*, pp. 33–36, IEEE, 2002.

[108] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 15, no. 11, pp. 1148–1161, 1993.

[109] M. Vatsa, S. Richa, and P. Gupta, "Comparison of iris recognition algorithms," in *Proceedings of International Conference on Intelligent Sensing and Information Processing*, pp. 354–358, IEEE, 2004.

[110] J. Deng and F. Lai, "Region-based template deformation and masking for eye feature extraction and description," *Pattern Recognition*, vol. 30, no. 3, pp. 403–419, 1997.

[111] K. Lam and H. Yan, "Locating and extracting the eye in human face images," *Pattern Recognition*, vol. 29, no. 5, pp. 771–779, 1996.

[112] Y. Wu, H. Liu, and H. Zha, "A new method of human eyelids detection based on deformable templates," in *Proceedings of IEEE international Conference on Systems, Man and Cybernetics (SMC'04)*, (Hague, Netherlands), Oct 2004.

[113] J. Zhu and J.Yang, "Subpixel eye gaze tracking," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, 2002.

[114] C. A. Perez, A. Palma, C. A. Holzmann, and C. Pena, "Face and eye tracking algorithm based on digital image processing," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, (Tucson, Arizona, USA), pp. 1178–1183, Oct 2001.

[115] X. Xie, R. Sudhakar, and H.Zhuang, "A cascaded scheme for eye tracking and head movement compensation," *IEEE Transactions on Systems, Man and Cybernetics- Part A: Systems and Humans*, vol. 28, no. 4, pp. 487–490, 1998.

[116] X. Xie, R. Sudhakar, and H. Zhuang, "Real-time eye feature tracking from a video image sequence using kalman filter," *IEEE Transactions on Systems, Man, Cybernetics*, vol. 25, pp. 1568–1576, Dec 1995.

[117] X. Xie, R. Sudhakar, and H. Zhuang, "On improving eye feature extraction using deformable templates," *Pattern Recognition*, vol. 27, pp. 791–799, 1994.

[118] Y. Ebisawa and S. Satoh, "Effectiveness of pupil area detection technique using two light sources and image difference method," in *Proceedings of International conference on IEEE Engineering in Medicine and Biology Society*, pp. 1268–1269, IEEE, Oct 1993.

[119] C. H. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," *Journal of Image and vision Computing*, vol. 18, pp. 331–335, March 2000.

[120] D. W. Hansen, J. P. Hansen, M. Nielsen, and A. S. Johansen, "Eye typing using markov and active appearance models," in *Proceeding of the Sixth IEEE Workshop on Applications of Computer Vision*, pp. 132–136, IEEE, Dec 2002.

[121] D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head," in *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 451–458, IEEE, June 2003.

[122] A. M. Al-Qayedi and A. F. Clark, "Constant-rate eye tracking and animation for model-based-coded video," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 6, pp. 2353–2356, IEEE, June 2000.

[123] A. Kapoor and R. Picard, "Real-time, fully automatic upper facial feature tracking," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR02)*, pp. 8–13, IEEE, May 2002.

[124] F. Bourel and C.C.Chibelushi, "Robust facial feature tracking," in *Proceedings of the 11th British Machine Vision Conference (BMVC2000)*, (Bristol, UK), British Machine Vision, September 2000.

[125] Y. Sheng, A. Sadka, and A. Kondoz, "Automatic 3d face synthesis from a single video frame," in *Research Excellence Awards Competition2003*, University of Surrey, 2003.

[126] T. Goto, S. Kshirsagar, and N. Thalmann, "Automatic face cloning and animation using real-time facial feature tracking and speech acquisition," *Signal Processing Magazine*, vol. 18, pp. 17–25, May 2001.

[127] Q. Chen, W. Cham, and H. Tsui, "A method for estimating and accurately extracting the eyebrow in human face image," in *Proceedings of International Conference on Image Processing (ICIP)*, vol. III, pp. 793–796, IEEE, 2002.

[128] J. Y. Wang and G. D. Su, "The research of chin contour in fronto-parallel images," in *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, (Xi'an), pp. 2814–2819, IEEE, November 2003.

[129] M. Kampmann, "Estimation of the chin and cheek contours for precise face model adaptation," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 26–29, IEEE, Oct 1997.

[130] X. B. Li and N. Roeder, "Face contour extraction from front-view images," *Pattern Recognition*, vol. 28, no. 8, pp. 1167–1179, 1995.

[131] M. Hu, S. Worrall, A. H. Sadka, and A. M.Kondoz, "A fast and efficient chin detection method for 2-d scalable face model design," in *Proceedings of International Conference on Visual Information Engineering (VIE2003)*, pp. 121–124, IEE, July 2003.

[132] T. Goto, W. S. Lee, and N. M. Thalmann, "Facial feature extraction for quick 3-d face modelling," *Signal Processing: Image Communication*, vol. 17, pp. 243–259, 2002.

[133] Z. C. Liu, Z. Y. Zhang, C. Jacobs, and M. Cohen, "Rapid modelling of animated faces from video," *The journal of visualization and computer animation*, vol. 12, pp. 227–240, 2001.

[134] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 189–192, 1988.

[135] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge: MIT Press, 1993.

[136] Z. Zhang, "Motion and structure from two perspective views: from essential parameters to euclidean motion via fundamental matrix," *Journal of the Optical Society of America*, vol. 14, no. 11, pp. 2938–2950, 1977.

[137] Z. Liu, "A fully automatic system to model faces from a single image," Tech. Rep. MSR-TR-2003-55, Microsoft Research, August 2003.

[138] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, "Synthesizing realistic facial expressions from photographs," in *Computer Graphics, Annual Conference Series, Siggraph*, pp. 75–84, 1998.

[139] S. H. Luo and R. W. King, "Automatic human face modeling in model-based facial image coding," in *Proceedings of Australian New Zealand Conference on Intelligent Information Systems*, (Adelaide), pp. 174–177, IEEE, November 1996.

[140] S. H. Luo, *Speech-enhanced model-based video facial image coding*. PhD thesis, University of Sydney, 1995.

[141] http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/.

[142] International Standard Organization, *JTC 1/SC 24; ISO standards*, 1 ed., Aug 2003.

[143] A. M. Tekalp and J. Ostermann, "face and 2-d mesh animation in mpeg-4," *Journal of Signal Processing on Image Communication*, vol. 15, pp. 387–421, 2000.

[144] E. Petajan, "The communication of virtual human faces using mpeg-4," in *Proceedings of International Symposium on Circuits and Systems*, vol. I, (Geneva), pp. 307–310, IEEE, May 2000.

[145] F. Pereira, "Mpeg-4: Why, what, how and when?," *Journal of Signal Processing on Image Communication*, vol. 15, pp. 271–279, 2000.

[146] M. Rydfalk, "Candide, a parameterized face," Tech. Rep. LiTH-ISY-I-866, Department of Electrical Engineering, Linkoping University, Sweden, 1987.

[147] B. Welsh, *Model-Based Coding of Images*. PhD thesis, British Telecom Research Lab, 1991.

[148] http://www.web3d.org/.

[149] http://www.cyberware.com/.

[150] http://www.opengl.org/.

[151] N. Eveno, A. Caplier, and P. Y. Coulon, "A new color transformation for lips segmentation," in *Proceedings of Multimedia Signal Processing IEEE Fourth Workshop*, pp. 3–8, 2001.

[152] http://mathworld.wolfram.com/Euler LagrangeDifferentialEquation.html.

[153] http://www.owlnet.rice.edu/ elec301/Projects99/faces/images.html.

[154] T. P. Weldon and W. E. Higgins, "Design of multiple gabor filters for texture segmentation," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, (Atlanta, GA), pp. 2243–2246, IEEE, May 1996.

[155] J. G. Zhang, T. Tan, and L. Ma, "Invariant texture segmentation via circular gabor filter," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 901–904, IEEE, Aug 2002.

[156] A. Laine and J. Fan, "Texture classification by wavelet packet signature," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1186–1191, Nov 1993.

[157] P. Ekman, "Mett-micro expression training tool- a cd rom," 2003.

[158] P. Kelly, E. Cooke, N. O'Connor, and A. Smeaton, "Detecting shadows and low-lying objects in indoor and outdoor scenes using homographies," in *Proceedings of visual information engineering (VIE),IEE*, (Glasgow, UK), pp. 393–400, IEE, April 2005.

[159] K. Tomiyama, M. Katayama, Y. Orihara, and Y. Iwadate, "Arbitrary viewpoint images for performances of japanese traditional art," in *Proceedings of the European Conference on Visual Media Production, IEE*, (London, UK), pp. 68–75, IEE, Nov 2005.