1

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

MACHINE IDENTIFICATION OF FACIAL IMAGES

by

Ian J. H. MacLaren, B.Sc.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Science

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
December 21, 1989

© copyright 1989
Ian J. H. MacLaren

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-59049-1

Acceptance

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance of the thesis, "*Machine Identification of Facial Images*", submitted by Ian J.H. MacLaren in partial fulfillment of the requirements for the degree of Master of Science.

Thesis Supervisor

Chairman, Department of Systems
and Computer Engineering,
Carleton University

Abstract

This thesis investigates the feasibility of implementing a computer vision system designed to identify human facial images. Experimental techniques for extracting a facial image from a background scene are discussed, as are algorithms for locating boundaries and fiducial points within the facial region of an image. A structural model based on manually collected facial statistics was used to facilitate the automation of the feature extraction process. Geometric fiducial data obtained from an image was encoded in the form of a n -dimensional feature vector. Feature vectors were then used for classification of individuals or for identification purposes.

Acknowledgements

I would like to express my appreciation to my thesis supervisor, Professor H.M. Schwartz for his guidance and supervision during the course of this research. As well, I wish to thank Dr. David C. Coll, who provided many valuable comments and suggestions.

I would also like to thank my colleague Mr. Yiyan Wu for his encouragement and for many enlightening discussions during various stages of this work. I am also grateful to numerous faculty members and graduate students in the department, who provided constructive criticism and feedback.

Finally, I would like to dedicate this thesis to my parents, Roy and Lee MacLaren and to Bardolph MacLaren.

Table of Contents

Acceptance	ii
Abstract	iii
Acknowledgements	iv
List of Figures	vii
CHAPTER 1	INTRODUCTION 1
1.1	Background 1
1.2	Objectives and Methodology 2
1.3	Thesis Outline 3
1.4	Symbol Dictionary 3
CHAPTER 2	A REVIEW OF FACIAL RECOGNITION TECHNIQUES . . 5
2.1	The Facial Identification Paradigm 5
2.2	Profile Recognition 6
2.3	Facial Recognition Techniques 11
CHAPTER 3	FEATURE EXTRACTION AND ISOLATION
	ALGORITHMS 19
3.1	Overview 19
3.2	Local Environment & Processing Constraints 21
3.3	System Calibration 22
3.4	Extraction of The Subject From The Background Scene 24
3.4.1	Subject Positioning & Image Capture 25
3.4.2	The Image Subtraction Step 26
3.4.3	Occluding Edge Detection 27
3.5	Orientation Analysis 36
3.6	Isolation of The Head and Facial Boundaries 38
3.6.1	Definition of The Primary Facial Boundaries and Location of The Neck & Shoulders 39
3.6.2	Placement of The Facial Window 42
3.7	Locating The Subject's Eyes 44
3.8	Locating the Nose and Mouth 51
3.9	Manual Point Collection and Feature Projection 59

CHAPTER 4	FEATURE CLASSIFICATION AND RECOGNITION	63
4.1	Facial Measurements and Ratios	63
4.1.1	Automated Feature Extraction	64
4.1.2	Manual Feature Extraction	65
4.2	Encoding and Storage of Feature Vectors	70
4.3	Classification & Recognition Methodology	79
4.4	Summary of The Facial Classification & Identification Process . . .	84
4.4.1	Automated Feature Extraction & Identification . . .	84
4.4.2	Manual Feature Extraction & Automated Identification	86
CHAPTER 5	DISCUSSION	90
5.1	System Performance	90
5.1.1	Overview	90
5.1.2	Selection of Test Images	92
5.1.3	Selection of Test Feature Vectors	93
5.1.4	Identification Test Results	103
5.1.5	Processing Speed	105
5.2	Facial Shape Recovery	107
5.2.1	Light Source Orientation	107
5.2.2	Reflectance Map Experimentation	110
CHAPTER 6	CONCLUSIONS	115
References		118
Appendix A	Processing Environment	120
Appendix B	Source FACEMAP Point Data	123
Appendix C	Source Listing of REC.C	129

List of Figures

Figure 2.1:	A Description of features used by Harmon et al.	8
Figure 2.2:	The profile and 11 fiducial points used by Harmon et al.	9
Figure 2.3:	A feature scorecard for the eyebrows.	11
Figure 2.4:	The registration diagram used by Cannon et al in locating features on the subject's head.	15
Figure 2.5:	The fiducial features used by Cannon et al.	16
Figure 2.7:	Pseudocode for Cannon's eye-fitting mechanism	16
Figure 2.6:	A diagram showing fiducial features used by Cannon et al.	17
Figure 3.8:	The facial points located automatically by the automated feature extraction algorithm.	20
Figure 3.9:	Overview of the procedure used for automatic determination of an image subtraction threshold.	24
Figure 3.10:	An ideal view of the subject during the image acquisition phase.	26
Figure 3.11:	Some commonly used edge detection and filtering templates.	28
Figure 3.12:	Execution times for several edge isolation and detection mechanisms.	29
Figure 3.13:	Search directions for the left and right occluding edge vectors.	30
Figure 3.14:	The scanline based edge detection mechanism.	32

Figure 3.15:	A three-dimensional intensity mesh of the area including and surrounding the face of Subject1.	35
Figure 3.16:	A three-dimensional intensity mesh of the area including and surrounding the face of Subject2.	35
Figure 3.17:	Calculation and placement of the symmetry equation and centre vector.	37
Figure 3.18:	Overview of the recursive two-dimensional orientation correction algorithm.	38
Figure 3.19:	Location of the initial head boundaries.	41
Figure 3.20:	The search for the neck and shoulder boundary.	41
Figure 3.21:	Final placement of the head boundaries and head window.	42
Figure 3.22:	Placement and adjustment of the facial window.	44
Figure 3.23:	Placement of the left and right eye windows.	45
Figure 3.24:	The pixel clustering algorithm used to locate the pupil of a subject's eye.	48
Figure 3.25:	Horizontal scans taken from the area surrounding the eyes of Subject1.	49
Figure 3.26:	Horizontal scans taken from the area surrounding the eyes of Subject2.	49
Figure 3.27:	A Three-dimensional intensity mesh of the area surrounding the right eye of Subject1.	50
Figure 3.28:	A three-dimensional intensity mesh of the area surrounding the right eye of Subject2.	50
Figure 3.29:	Placement of the nose and mouth windows.	52

Figure 3.30:	The orientation of scanline sets used in the scanline graphs.	53
Figure 3.31:	Horizontal scans taken from the area surrounding the nose of Subject1.	54
Figure 3.32:	Horizontal scans taken from the area surrounding the nose of Subject2.	54
Figure 3.33:	A Three-dimensional intensity mesh of the area surrounding the nose of Subject1.	55
Figure 3.34:	A three-dimensional intensity mesh of the area surrounding the nose of Subject2.	55
Figure 3.35:	Horizontal scans taken from the area surrounding the mouth of Subject1.	56
Figure 3.36:	Horizontal scans taken from the area surrounding the mouth of Subject2.	56
Figure 3.37:	A Three-dimensional intensity mesh of the area surrounding the mouth of Subject1.	57
Figure 3.38:	A three-dimensional intensity mesh of the area surrounding the mouth of Subject2.	57
Figure 3.39:	Vertical scans taken from the area surrounding the nose of Subject1.	58
Figure 3.40:	Vertical scans taken from the area surrounding the nose of Subject2.	58
Figure 3.41:	The eleven eleven facial points located manually (for use with 9 & 4 feature classification algorithms).	59
Figure 3.42:	Feature windows and points used in the projection of FaceMap data.	61

Figure 3.43:	The projection of manually collected FaceMap data. (used by the feature projection algorithm and 4 and 9 feature classification procedures)	62
Figure 4.44:	A description of features composed of distance ratios used by the automated feature classification algorithm to obtain 3 features.	65
Figure 4.45:	A description of features composed of distance ratios used by the manual feature classification algorithm to obtain 9 features.. . . .	66
Figure 4.46:	A description of features composed of distance ratios used by the manual feature classification algorithm to obtain 4 features.	67
Figure 4.47:	Distance ratios used by the earliest version of the automated feature classification algorithm to obtain 3 features.	68
Figure 4.48:	Distance ratios used by the final version of the automated feature classification algorithm to obtain 3 features.	68
Figure 4.49:	Distance ratios used by the manual feature classification algorithm to obtain 9 features.	69
Figure 4.50:	Distance ratios used by the manual feature classification algorithm to obtain 4 features.	69
Figure 4.51:	Automated feature extraction using 3 features: raw feature data.	72
Figure 4.52:	Automated feature extraction using 3 features: frequency distributions for the 1'st and 2'nd features.	72
Figure 4.53:	Automated feature extraction using 3 features: frequency distribution for the 3'rd feature.	73
Figure 4.54:	Manual feature extraction using 9 features: raw feature data.	74
Figure 4.55:	Manual feature extraction using 9 features: frequency distributions for the 1'st and 2'nd features.	74

Figure 4.56:	Manual feature extraction using 9 features: frequency distributions for the 3'rd and 4'th features.	75
Figure 4.57:	Manual feature extraction using 9 features: frequency distributions for the 5'th and 6'th features.	75
Figure 4.58:	Manual feature extraction using 9 features: frequency distributions for the 7'th, 8'th and 9'th features.	76
Figure 4.59:	Manual feature extraction using 4 features: raw feature data.	77
Figure 4.60:	Manual feature extraction using 4 features: frequency distributions for the 1'st and 2'nd features.	77
Figure 4.61:	Manual feature extraction using 4 features: frequency distributions for the 3'rd and 4'th features.	78
Figure 4.62:	Automated feature extraction using 3 features and 28 feature vectors: Raw Euclidian distance scores (top) and corresponding frequency distribution (bottom).	82
Figure 4.63:	Manual feature extraction using 9 features and 16 feature vectors: Raw Euclidian distance scores (top) and corresponding frequency distribution (bottom).	82
Figure 4.64:	Manual feature extraction using 4 features and 16 feature vectors: Raw Euclidian distance scores (top) and corresponding frequency distribution (bottom).	83
Figure 4.65:	Overview of the automated facial classification and recognition algorithm.	87
Figure 4.66:	Overview of automated facial classification and recognition algorithm (continued).	88
Figure 4.67:	Overview of manual training and recognition algorithm.	89

Figure 5.68:	Feature vector data (3 features) used by the fully automated identification algorithm (top), and sets of 4 test feature vectors for Subject1 and Subject2.	94
Figure 5.69:	Feature vector data (9 features) used by the partially automated identification algorithm (top), and sets of 4 test feature vectors for Subject1 and Subject2.	95
Figure 5.70:	Feature vector data (4 features) used by the partially automated identification algorithm (top), and sets of 4 test feature vectors for Subject1 and Subject2.	96
Figure 5.71:	Subject1: recognition scores for 4 test feature vectors, ranked against 28 known vectors, using automated extraction of 3 features. (top – Test Score1, 2nd from top – Test Score2, 3rd from top – Test Score3, bottom – Test Score4)	97
Figure 5.72:	Subject1: recognition scores for 4 test feature vectors, ranked against 16 known vectors, using manual extraction of 9 features. (top – Test Score1, 2nd from top – Test Score2, 3rd from top – Test Score3, bottom – Test Score4)	98
Figure 5.73:	Subject1: recognition scores for 4 test feature vectors, ranked against 16 known vectors, using manual extraction of 4 features. (top – Test Score1, 2nd from top – Test Score2, 3rd from top – Test Score3, bottom – Test Score4)	99
Figure 5.74:	Subject2: recognition scores for 4 test feature vectors, ranked against 28 known vectors, using automated extraction of 3 features. (top – Test Score1, 2nd from top – Test Score2, 3rd from top – Test Score3, bottom – Test Score4)	100
Figure 5.75:	Subject2: recognition scores for 4 test feature vectors, ranked against 16 known vectors, using manual extraction of 9 features. (top – Test Score1, 2nd from top – Test Score2, 3rd from top – Test Score3, bottom – Test Score4)	101

Figure 5.76:	Subject2: recognition scores for 4 test feature vectors, ranked against 16 known vectors, using manual extraction of 4 features. (top – Test Score1, 2nd from top – Test Score2, 3rd from top – Test Score3, bottom – Test Score4)	102
Figure 5.77:	A summary of recognition test scores based on 3 sets of 4 test feature vectors (from the 4 test images) for both Subject1 and Subject2.	103
Figure 5.78:	A three-dimensional intensity mesh of the ping-pong ball image used for reflectance map experimentation.	113
Figure 5.79:	Intensity contours and light source points found on the ping-pong ball.	114
Figure 5.80:	A reflectance map represented as a lookup table, showing a possible single point light source.	114
Figure A.82:	Matrox PIP-512 pixel transfer rates.	121
Figure A.81:	Overview of the system hardware configuration.	122

Chapter 1

INTRODUCTION

1.1 Background

This thesis explores the issues surrounding the implementation of a computer vision system designed to identify images of human faces. Although such a system has never been constructed for use in a commercial or industrial setting, the capability of automatic facial identification by machine is one which would have substantial practical application in areas ranging from law enforcement and physical security access systems to anthropological research. Previous studies undertaken in this area by L.D Harmon in [1] and S.R Cannon *et al* in [2] yielded profile and facial identification rates of up to 96% using experimentally developed and relatively efficient but modest algorithms. However, the theoretical basis for much of the research in the area appears to be deficient. Regardless, the problem is very challenging. Given an image of a human face, how can a computer vision system be made to extract the facial component from the image, isolate various fiducial features and compare these features with those stored in a feature database for identification purposes? In 1978 Harmon and Hunt stated in [3] that “machine manipulation of information that sufficiently represents a pictorial scene as complex as a human face is generally beyond the present state-of-the-art.” More than a decade later, this statement would still appear to be true. Nonetheless, this thesis demonstrates that highly specialized techniques may be used to obtain a reasonable solution for the facial identification problem.

1.2 Objectives and Methodology

The overall objective of this research was to demonstrate the feasibility of implementing a facial identification system. This required the development and evaluation of algorithms which could not only take advantage of the geometric structure and symmetrical aspects of a face, but could also extract information from a facial image and subsequently convert this information into a form suitable for identification purposes. In the absence of any machine-based facial identification models, techniques were initially developed to support a fiducial model based on manually collected facial statistics. In turn, the results from this process were used to develop automated feature extraction algorithms and to test the identification mechanisms.

Whenever practically possible, techniques used during the course of this research followed existing methods utilized in the literature. On the other hand, the facial identification paradigm was found to be highly specialized and many of the algorithms and techniques developed here were of an experimental nature. Furthermore, there was a complexity versus speed trade-off issue that arose due to the possibility that an operational system of this nature would have real-time constraints placed upon it, so that many of the individual techniques developed were relatively simple, but very efficient. Relatively complex algorithms were only chosen if they provided significant benefits over a more efficient one. By using a numerous assortment of simple but fast algorithms, instead of a few relatively complex ones, it was easier to evaluate the merits and drawbacks of a particular approach. This, in turn, facilitated the construction of several experimental models. In brief, the methodology used coincided with that used by Harmon *et al* who summarized in [3]: “The detections and decisions described ... are *ad hoc* evolutions of machine algorithms; they are suggested by human performance, but are tailored to and constrained by rather uncomplicated computer techniques.”

The original contributions made by this research are as follows: First, it provides a geometric and structural facial model suitable for machine-based identification purposes,

secondly, it introduces a collection of feature extraction and classification algorithms, and thirdly, it demonstrates the feasibility of implementing both a partially and fully automated facial identification system in a relatively unconstrained setting.

1.3 Thesis Outline

The thesis material is organized in the following manner: Chapter 2 summarizes previous research in and related to the area of facial identification, with focus on two studies of particular relevance. The 3rd Chapter describes the algorithms and procedures used for both fully and partially automated extraction of fiducial data from facial images. Chapter 4 describes the means by which fiducial data were transformed for use by the classification and recognition algorithms, with particular attention paid to the identification mechanisms. The 5th Chapter presents a detailed set of test results and covers several issues of immediate interest, including a discussion of future research directions. Concluding remarks are contained in Chapter 6.

Appendix A describes the processing environment used, and provides an overview of the software implementation and hardware details. Appendix B contains a fiducial data set used for feature projection¹, and Appendix C contains a small but interesting code fragment belonging to the primary recognition module.

1.4 Symbol Dictionary

A description for many of the commonly used symbols and abbreviations follows. Although use of these symbols is for the most part consistent, there are several instances

¹ of particular relevance to the discussion in Chapter 3.

in which they describe or represent other items. In such cases, a local description of the symbol is provided instead.

Functions

λ : scanline edge detection function
 γ : vector smoothing (local average) function
 δ : screen-normalized Euclidian distance function
 ξ : n-dimensional Euclidian distance function
 w : scalar boundary function
 ρ : cartesian point function

Vectors & Matrices

Ω : occluding edge vectors for the head
 Ψ : centre vector
 ψ : symmetry vector
 Υ : shoulder distance vector
 Θ : occluding edge vectors for a ping-pong ball
 I : general purpose image matrix
 F : foreground image matrix
 B : background image matrix
 S : subtracted image matrix

Scalars

τ : edge detection or image subtraction threshold
 η : distance between the pupils of the eyes
 φ : screen aspect ratio
(boundaries)
 tb : top boundary
 bb : bottom boundary
 lb : left boundary
 rb : right boundary
 cb : centre boundary
 mb : middle boundary
 $mline$: vertical position of the mouth
 $eyeline$: vertical position of line between eyes
 $neck$: vertical position of the neck

Test Subjects

Subject1: test subject #1
Subject2: test subject #2

Throughout the discussion are diagrammatical and graphical presentations which make reference to *Subject1* and-or *Subject2*. Each of these symbols correspond to one of two individuals used for testing purposes.

Chapter 2

A REVIEW OF FACIAL RECOGNITION TECHNIQUES

2.1 The Facial Identification Paradigm

The area of facial identification is indeed an interdisciplinary one, although the vast majority of past research has been closely related to the the area of human perception and psychology. Bradshaw *et al* in [4] describe the human face as a “complex visual pattern.” Apparently, this complexity is primarily due to the abundance of fiducial features as well as their large number of possible variations. The aim of Bradshaw’s paper was to determine the speed at which people could recognize facial images based on Identikit constructions. However, issues concerning parallel and serial processing during facial identification were also examined and levels of difficulty were measured when the faces were presented in a variety of unexpected formats. Bradshaw suggested facial identification by humans becomes a parallel operation when the face is familiar, although when the face is unfamiliar or presented in an unusual way a serial approach is taken instead. For example, Bradshaw indicated upside-down faces were more difficult to recognize than right-side-up faces, and that people usually chose one of two approaches to identification when faced with this dilemma. Subjects tested by Bradshaw would usually search for a distinguishing feature on the face or attempt to gain an overall impression based on several features. A similar study carried out by Goldstein *et al* in [5] examined the ability of children to recognize photographs of human faces as well as

exploring the question of which facial features were most useful for identification.

Unfortunately, having knowledge of the means by which humans recognize faces may be of little use when designing a suitably equivalent machine implementation, as the two identification paradigms are quite opposite. Machines are not known to be particularly “impressionable.” Harmon points out in [3] that “humans are very good at differentiating subtle differences in patterns, regardless of shape deviations, whereas computers are not. Computers are better at determining and using qualitative measures, which are not necessarily “patterns” for humans.” In the context of the facial identification paradigm described in this thesis, the goals of the machine were twofold. The first was to locate a subset of facial features common to all humans, whereas the second was to quantify these features with the greatest possible precision.

The following sections focus in detail upon two areas of particular relevance. The first area deals with profile recognition research carried out by Harmon and his colleagues. Although Harmon never managed to implement a machine based facial identification system, his work in profile recognition was particularly valuable and enlightening. It turned out many of the techniques used by Harmon could be applied to the facial identification problem. Research carried out in 1986 by Cannon *et al* in [2] is of immediate interest because it would appear to represent the only *published* description of a functional automated facial recognition system. The system developed by Cannon is comparable in several ways to the one described in subsequent chapters.

2.2 Profile Recognition

Research undertaken by Harmon *et al* in [3] and [1] dealt with the automatic recognition of facial profiles. The first study used 256 subjects, and 11 features, whereas the second study used 112 subjects and 17 features. The second study simply extended and refined the techniques used in the first one. Subjects were photographed against a

background of high contrast (eg. silhouettes) a total of 4 times, at roughly one week intervals. These photographs were then processed by a scanner and digitized. For each profile, the spatial resolution of the digitized picture was fairly low; approximately 300 points on each profile were obtained. Harmon felt the low resolution and small amounts of measurement error would not pose any significant problems in terms of the recognition accuracy of his system. As well, by using profile “silhouettes”, the representation of data was relatively straightforward; a binary matrix in this case.

Once each photograph had been digitized, 11 points on the facial profile were located automatically by a minicomputer. These points were used to define various geometric fiducial features composed of distances, angles and areas. The generic profile and the points used in Harmon's first study are shown in Figure 2.2. Similarly, Figure 2.1² describes the set of features used, also corresponding to the fiducial profile points shown in Figure 2.2. These geometric features were eventually encoded in the form of an 11-dimensional feature vector for identification purposes. Pairwise differences between known and unknown feature vectors were defined using 11-dimensional normalized Euclidian distances.

A set of three profiles obtained for each individual were used for training purposes, while a fourth was treated as an unknown so it could be used for testing. Thus for the second study, a total of 363 known profiles were maintained by the system: a profile for each of 3 sittings $\times$ 121 people. This information was maintained in what Harmon referred to as a “primary” or P file. An additional file named the “reference” or R file contained a mean feature vector for each known individual which was obtained by averaging the vectors from the three sittings. This operation was defined by

$$r_{ij} = \frac{1}{3} \sum_{k=1}^3 r_{ijk} \quad (1)$$

where the indices i, j, k correspond to the feature, individual, and sitting number respectively. Euclidian distances were computed for an unknown feature vector p_i against

² Reproduced from [3]

Feature #	Feature Description
1	Nose Protusion, P. The distance of point 1 from the baseline 2-3.
2	Area. The absolute magnitude of the areas framed by the profile trace and the line 2-3.
3	Angle. Measured between lines 1-2 and 2-3 (eg. angle 123).
4	WIGU. "Wiggle up", the average of the inverse radius of curvature at each sample point between fiducial marks 1 and 3.
5	WIGD. "Wiggle down", as above, between fiducial marks 1 and 3.
6	R12. The length of the line joining points 1 and 2
7	R14. The length of the line joining points 1 and 4.
8	R15. The length of the line joining points 1 and 5.
9	R16. The length of the line joining points 1 and 6.
10	R18. The length of the line joining points 1 and 8.
11	R79. The length of the line joining points 7 and 9.

Figure 2.1 A Description of features used by *Harmon et al.*

all known feature vectors in R . Thus, the Euclidian distance D between the set of unknown feature vectors P and known feature vectors R was defined as

$$D(P, R) = \sqrt{\left\{ \sum_{i=1}^{11} \left[\frac{p_i - r_i}{s_i} \right]^2 \right\}} \quad (2)$$

where the normalization factor, s , represented the standard deviation for each feature. Equation 3 shows how s is obtained for each feature. During the earlier study, the

Figure 2.2 The profile and 11 fiducial points used by Harmon *et al.*

feature variance was used as the normalization factor instead.

$$s_i = \sqrt{\left\{ \frac{1}{121} \sum_{j=1}^{121} \left[\frac{1}{2} \sum_{k=1}^3 (r_{ijk} - r_{ij})^2 \right] \right\}} \quad (3)$$

The resultant distances were then ranked and sorted in ascending order, and the vector in R closest to the unknown vector p_i was chosen as the vector corresponding to and identifying the individual. This final selection process was then defined as

$$D(P_j, R_j) = \min(D(P_j, R_i)) \quad (4)$$

To improve performance³ and the accuracy of his system, Harmon chose to implement a type of feature thresholding scheme. For example, if the feature represented by the distance from the nose-tip to the eye (points 1 and 4 in Figure 2.2) was not within a pre-assigned range then the R feature vector corresponding to that individual could be excluded from further processing and from consideration as a possible match. This threshold was expressed as a feature range which Harmon referred to as a "window". The width of each feature window was defined as the "maximum absolute percentage deviation" from the corresponding average vector component. Thus, for the vector R to qualify as a possible match, each vector component had to fall within the window as defined by

$$w_{ij} = \max \left| \frac{|\mu_{ij} - r_{ijk}|}{\left(\frac{\mu_{ij} + r_{ijk}}{2} \right)} \right|, \quad \text{where } k = \{1, 2, 3\} \quad (5)$$

where $\mu_{ij} = \frac{\sum_{k=1}^3 r_{ijk}}{3}$ refers to the the mean feature value over three sittings. Individuals who failed to meet the conditions in equation 6 were excluded from further processing and as a candidate for a possible match. For any P vector tested, the following thresholding

³ eg to reduce the number of Euclidian distance calculations required by the system during the recognition phase

constraints were applied:

$$w_{ij} \geq \left(\frac{|\mu_{ij} - p_i|}{\left(\frac{\mu_{ij} + p_i}{2} \right)} \right) \quad (6)$$

Harmon managed to significantly improve the recognition accuracy of his system by engaging this type of technique, particularly in the second study, by using a set of 17 features which had been selected after very lengthy and detailed analysis. He demonstrated features which were highly correlated with each other should not be used, as a serious degradation in recognition accuracy would result. In the course of testing his 17-feature [1] implementation, Harmon managed to attain a 96% recognition rate.

2.3 Facial Recognition Techniques

Harmon's earliest research in the area of identification can be found in [6] and [7]. This work dealt with facial identification by humans, and laid the foundation for his research into automated profile recognition several years later. A total of 256 subjects were used. For each individual, three photographs were taken; the first a full frontal shot, the second a three-quarter profile, and the third a full profile. The photographs for the entire population were qualitatively assessed by a human "jury", to produce a rating for each of the 22 facial features used. For example, in Figure 2.3, an assessment of "medium" for a subject's eyebrows would have been rated a three.

EYEBROWS	1 Thin	2	③ Medium	4	5 Bushy
-----------------	-----------	---	-------------	---	------------

Figure 2.3 A feature scorecard for the eyebrows.

These qualitative feature values were manually encoded in the form of a feature vector, although the vector was not subject to identification processing by machine. For such tests, individuals were given a precise description of a person and a search for an

ideal candidate was conducted manually, based on the feature information produced by the jury. Tests were carried out on population sizes ranging from 64 to 255, and an average identification rate of about 50% was achieved using this technique. Although not automated, the process was impressive in its ability to match pairs of similar faces, as well as its ability to discern between dissimilar faces. Nonetheless, this study did suggest humans were not particularly well suited to facial identification using features alone, especially over a large population of individuals.

M. L. Rhodes in [8] employed an interactive natural-language-like parser to generate facial images based on Identi-Kit features. Although the dialogue was somewhat limited, a user would first describe a human face using English statements and the system would subsequently produce a composite facial image based on the description. Such an image would have likely been suitable as input for two of the partially automated identification algorithms described in later chapters. The language parser was capable of dealing with 11 facial features, each of which could be interactively adjusted for size, shape and location through the use of English statements. For the most part, this work represented a substantially improved human interface or front-end applicable to Harmon and Goldstein's [6] and [7] studies, although no automatic identification was performed. Rhodes also indicated that in the case of another composite kit⁴ named Photo-Fit, there existed over 12,000 million possible feature combinations.

It is interesting to note that all existing techniques related to machine based facial identification have taken a geometric or decision theoretic approach. Furthermore, it is surprising a structural or syntactic approach⁵ has not been used to date. This would amount to the inverse of Rhodes facial construction techniques, although one inhibiting factor would be the difficulty in defining an appropriate set of pattern primitives for a human face. However, commercial fingerprint identification systems have used a syntactic approach with a great degree of success. Naturally, this raises the question: are human faces "fingerprintable"?

⁴ Similar to Identi-Kit.

⁵ eg. one based on the theory of Automata.

The study conducted by Cannon *et al* in [2] used a total of 50 known subjects, 25 strangers and was fully automated. The environment in which all subject images were acquired was carefully controlled and constant. Two light sources located at approximately head level were maintained at exactly 45 degree angles to a neutral grey background. During the subject acquisition phase, individuals were first required to stand in front of the background, approximately 5 feet away from the a video camera. The camera which was adjusted to obtain a reasonable view of the subject's face, and a snapshot containing the subject's head an shoulders was acquired. This arrangement was apparently designed to ensure the image area occupied by the subject's head and shoulders at least half of the entire frame. As in all of Harmon's studies, no beards, moustaches, glasses, or any other foreign objects were allowed. In preparing an image for processing, all images were normalized or scaled to a predetermined background level.

Cannon used the term "registration" to describe a 6-part process in which various algorithms were used to locate facial features and boundaries. Prior to invoking the registration process on a test image, the following assumptions were made:

- I. The top of the subject's head was clearly visible.
- II. The subject's head was approximately horizontally centered.

Before the algorithm could proceed, or at least at a very early stage the registration process, a background marking needed to be located as their system used physical units of measure (eg. *cm*) when searching for and predicting the location of features. This "fiducial" background mark was located roughly at head level and was apparently used to determine the height of the individual. Consequently, it is expected that location of this mark was carried out during the first step of the registration process. The various registration steps are summarized below and illustrated in Figure 2.4.⁶ Registration proceeded in the following manner:

⁶ Both the registration and feature diagrams (Figure 2.6) have been reproduced as closely as possible from [2].

- **[Step1]** The top of the individual's head was isolated by conducting a vertical search downwards from the top image boundary until an edge or "a significant intensity gradient" was found. If the boundary was not located in the top half of the frame, the image was rejected.
- **[Step2]** A calculation apparently based on the height of the individual was first used to predict the vertical location of an area between the subject's shoulders and ears. The outer facial boundaries were then isolated by searching inward from the left and right at the predicted vertical location for an edge. Evidently, if the separation between the left and right sides of the face was not in the 10–17cm range, the image was rejected.
- **[Step3]** Based on results obtained in steps 1 and 2, the location of the right⁷ cheekbone was established by predicting the location of a "box" surrounding the cheekbone. After analysis of this region, the 3×3 region of brightest pixels was chosen to correspond to the cheekbone. Cannon indicated this step (and hence the registration procedure) would fail if "a portion of the eye or bridge of the nose" fell within the search region.
- **[Step4]** Based on the cheekbone coordinates, a search region for the subject's right⁸ eye was predicted. The entire pupil of the eye had to be contained in this region for the fifth step to function correctly.
- **[Step5]** A search for the pupil of the right eye (bounded by the region established in step 5) was conducted. At least 6 intensity minimums were located, and a solution for each point was taken to include a region containing "all pixels showing a negative gradient towards the minimum." Solution areas intersecting with the boundaries of the search area were not considered. The location of the pupil was predicted and the final location was taken as the local minimum of the region closest to the predicted point.
- **[Step6]** A small region was defined on the left side of the face based on the coordinates of the right eye. The location of the left pupil corresponded to the coordinates of the intensity minimum within this region.

⁷ Left cheek-bone as seen by the viewer.

⁸ Left eye as seen by the viewer.

Once the registration process had been successfully completed, a set of 5 primary features were obtained. These features are itemized in Figure 2.5, and illustrated in Figure 2.6.

Figure 2.4 The registration diagram used by Cannon *et al* in locating features on the subject's head.

For each known individual, a set of 3 to 8 training images were maintained. As well, a test image for each of the 50 known individuals and 25 strangers were placed in a test set. A thresholding technique based on features 1–5 was used to eliminate individuals from the test set before submission to the identification procedure. This technique appeared to be somewhat analogous to Harmon's use of feature windows. For a test image to be considered for identification, various constraints based on height, hair intensity, forehead height, cheek intensity and eye intensity had to be satisfied.⁹ For the test images which met the selection criterion, identification was based on a least squares fit of a 5 cm^2 pixel matrix surrounding the left eye. This eye matrix was shifted about the origin by ± 3 pixels, to compensate for orientation problems and other errors occurring

⁹ Three thresholds were based on a grey scale range, and two were based on a range measured in *cm*.

Feature #	Feature Description
1	Height. (feature #1 in diagram)
2	Hair Intensity. (feature #2 in diagram)
3	Forehead Height. (feature #3 in diagram)
4	Cheek Intensity. (feature #4 in diagram)
5	Eye Intensity (feature #5 in diagram)
(6)	Left Eye Pixel Matrix.
(7)	Right Eye Pixel Matrix.

Figure 2.5 The fiducial features used by Cannon *et al.*

during the registration process. The pseudocode describing this procedure (Figure 2.7) is reproduced from [2].

```

FOR x := -3 to +3, Step 3 DO
  FOR x := -3 to +3, Step 3 DO
    BEGIN
      shift test matrix by (x,y);
      calculate least squares fit;
    END;
  
```

Figure 2.7 Pseudocode for Cannon's eye-fitting mechanism

Evidently the result corresponding to the best fit was retained. The confidence interval for the fit was computed, and if less than 95% level, the procedure was repeated using

Figure 2.6 A diagram showing fiducial features used by Cannon *et al.*

the other eye. If the confidence level for this eye was also less than 95%, the individual was deemed to be unknown. Cannon computed a recognition accuracy of 96% for his system, which was comparable to Harmon's optimal profile recognition rate.

Cannon suggested Harmon's techniques used "a significant amount of operator controlled posing" to obtain profiles, and that the profile measurements were subject to changes in facial expression. Both approaches would have been sensitive to orientation. However, Cannon's approach seemed largely dependent on lighting, although it was more independent of facial expression. It did, however, require a carefully controlled environment as well as one significant external clue — the fiducial background mark.

Harmon used an Optronics P-1700 drum scanner and a DEC PDP-11/45 minicomputer to carry out his research. A magnetic tape drive was used instead of a fixed disk drive. Initially, Harmon placed more emphasis on the development of algorithms instead of performance, so little of his source code was optimized. Each photo took approximately 1 minute to digitize on the drum scanner¹⁰. About 3 minutes were required to obtain

¹⁰ They were scanned in sets of ten.

the profile edge points and add the profile data to tape. A total of 6 seconds of CPU time were required for the isolation and extraction of profile features. Once a feature vector had been obtained, identification of a single "suspect" required about 14 seconds of search time. No performance information or statistics were provided by Cannon *et al* for their facial identification system, although the equipment¹¹ used was similar to the equipment used for the implementation described in the following chapters.

Although Harmon never used a video camera in any of his studies, he planned to construct a real-time implementation of his profile recognition system. Harmon also suggested the use of ultrasound, x-ray and infra-red scanners to deal with difficult problems such as hair. While his research was implemented on a small minicomputer, the methods he developed were well suited to real time implementation. Harmon indicated that [1] would conclude his research into recognition techniques based on geometric features. His agenda at the time included the development of an identification system based on human ear shapes.

The identification techniques described in this thesis employed a subset of the recognition methods used by Harmon in his profile identification work. These methods were particularly well suited for use in dealing with the facial identification problem as facial features were described using geometric distances. Furthermore, the techniques were effective, simple and efficient, making them adaptable for real-time implementation. Regrettably, neither Harmon or Cannon described their feature extraction algorithms in sufficient detail for their incorporation into the methods described in the following chapters.

¹¹ The imaging equipment used was a Microsonics frame grabber based in an IBM PC/XT microcomputer. The resolution of the frame grabber was 256x256, and 6 bits (64 gray levels) of image data were used.

Chapter 3

FEATURE EXTRACTION AND ISOLATION ALGORITHMS

3.1 Overview

During the course of this research, two methods were developed for identification of facial images. The first was fully automated in both its approach to feature extraction and to identification of an individual. Conversely, the second was only automated in the area of identification; the feature extraction procedures were carried out manually. In both the fully automated and partially automated instances, the rationale behind the algorithms and procedures described in this chapter was to produce a set of quantitative facial measurements or features which could be encoded in the form of a feature vector.

The aim of the feature extraction process was primarily to pinpoint various fiducial points and boundaries on a subject's face. Geometric distances between features could be used to represent a set of characteristics unique to an individual, which in turn could be encoded in the form of a feature vector. Once this had been accomplished, a feature vector could be compared with previously stored feature vectors for identification purposes, or added to a feature vector set of known individuals.

In a fully automated setting, the following procedures were undertaken to obtain the necessary facial information.

- I. Calibration of the System
- II. Acquisition and Isolation of the Subject
- III. Orientation Analysis and Correction
- IV. Isolation of The Head and Facial Boundaries
- V. Location of the Subject's Eyes
- VI. Location of the Subject's Nose and Mouth

The goal of the automated feature extraction algorithm was to automatically locate the 6 points on the subjects face, as shown in Figure 3.8. In the simpler case of a partially automated system, facial data was collected by a system operator using a graphics pointing device (eg. a mouse) to obtain the 11 facial points shown in Figure 3.41. In both the fully and partially automated circumstances, facial points were later used in the calculation of 3 to 9 geometric facial features for an individual. The feature extraction process is described in detail in the following sections.

Figure 3.8 The facial points located automatically by the automated feature extraction algorithm.

3.2 Local Environment & Processing Constraints

Facial images of subjects were acquired in a small room with lighting provided by overhead fluorescent lamps. The head of the subject was typically 2 to 6 feet away from the lens of the camera during the image acquisition phase and subjects were usually, but not necessarily, in a sitting position. Once the subject had been counselled on how to pose in an ideal manner, the imaging system was placed in a continuous frame grabbing mode. When the operator was satisfied with the subject's pose, a snapshot of the subject's head and shoulder was obtained. Subjects were asked to assume a natural facial expression and look directly at the camera lens with their mouth closed and eye glasses (if any) removed. Additionally, this implementation and those described in [2] and [1] did not allow beards and moustaches or other foreign objects such as hats. Acceptable facial images were of a clearly focused, well lit, unobstructed view of the subject's head. Several items pertaining to the image acquisition phase which were not precisely controllable included:

- I. Camera focus
- II. Camera aim
- III. Stability of camera base
- IV. Orientation of the subject
- V. Positioning of the backdrop

In summary, many operational parameters of the subject image acquisition phase could not be measured or controlled precisely. For example, orientation of the subject's face was quite arbitrary; individuals were simply told to look straight ahead at the camera lens.

It was not possible to precisely duplicate or maintain many of the operational conditions during the image acquisition phase. For experiments which made use of an arbitrary background, the subject's head often had to be positioned so the occluding edges of the head were in contrast with background objects. Furthermore, no use was made of external background clues, nor was any special equipment such as the following used:

- I. Background markings such as the fiducial mark used by Cannon *et al* in [2].

- II. Fixed or special lighting sources such as those used by Cannon *et al* and by Harmon *et al* in [1].
- III. A head positioning device which would help ensure proper and consistent orientation.

A Tandy 3000HD AT-compatible microcomputer equipped with an 80286 CPU, 80287 numeric co-processor and a Matrox PIP-512 frame grabber board was used for development of the facial identification programs. The dimensions of the frame buffer was 512×512 pixels ×8 bits deep. Earlier studies such as [2] suggested that as few as 6 bits/pixel would be sufficient for this type of quantitative facial image analysis. In all cases related to this research, 7 bits/pixel were used and the least significant image bitplane was used for graphic overlays. An input lookup table on the board was adjusted to map 7 bits of image data into the frame buffer. This arrangement allowed graphics to be written on top of the image without affecting the image data. The origin of the frame buffer coordinate system (0,0) was located at the top-left corner of the video monitor, while (512,480) was at the lower-right. On the vertical image axis, the frame memory between the 480th and 512th scan was not used, because the RS-170 video signal only provided 480 scans of usable image data. A more in-depth description of the processing environment is provided in Appendix A.

3.3 System Calibration

Two approaches were developed for automatically isolating the head and shoulders of a subject from the background component of an image. The first approach required the use of a constant background, whereas the second approach did not; it was designed to work with backgrounds of an arbitrary nature. However, the latter method also required an explicit calibration step before the subject acquisition phase. When an arbitrary background was used an image subtraction algorithm was used to calibrate the system. The aim was to provide the system with the capability of separating the

foreground portion of an image containing the subject from a background of an arbitrary nature. This involved acquiring an image of the background scene and a suitable image subtraction threshold which could subsequently be used for extracting the subject from the background. Starting with a value of zero, a suitable threshold τ , was automatically determined in advance by repeated subtraction of background frames against an initial background image. Simultaneously, τ was increased until all background noise had disappeared. Background "noise" was measured in terms of the frame-to-frame difference between successive background images. Before each incoming image I was averaged, it was reduced by retaining pixels in the following manner

$$I \equiv [I'_{x,y} = I_{4x,4y}], \text{ for } x = 0, \dots, 127, \text{ for } y = 0, \dots, 120 \quad (7)$$

to improve the performance of the procedure. The final value assigned to τ was also increased by $N \approx (15 \pm 5) \%$ to account for any measurement error and slight variations in background lighting. The procedure for determining τ is summarized in Figure 3.9. The calibration step as outlined in equation 8 was completed by acquiring several ($m \geq 2$) background images, averaging them and storing the averaged image in the background image matrix B .

$$B_{xy} = \frac{\sum_{n=1}^m I_{n(x,y)}}{m}, \text{ for } x, y = 0, \dots, 127 \quad (8)$$

The dimensions of I (and hence B) were chosen to be 128×480 , as opposed to 128×128 so that high resolution edge vector data could be obtained during the head and shoulder extraction phase (described in the following section). Consequently, incoming live images of the background were subject to the transformation:

$$B \equiv [I'_{x,y} = I_{4x,y}], \text{ for } x = 0, \dots, 127, \text{ for } y = 0, \dots, 479 \quad (9)$$

before they were averaged. Since the background scene needed to remain uniform and constant for the duration of the subject acquisition phase, it was also necessary beforehand to arrange for proper focusing and positioning of the video camera as well as adequate lighting.

Figure 3.9 Overview of the procedure used for automatic determination of an image subtraction threshold.

3.4 Extraction of The Subject From The Background Scene

Prior to locating any fiducial points or the measuring of fiducial distances, the automated feature extraction algorithm first required the occluding boundary data for the subjects head. These boundaries were used as reference points for all subsequent operations, and it was fundamental to the success of the automated extraction process that these edges be found consistently and with a high degree of accuracy. The outcome of this process indirectly resulted in separation of the image into the following distinct regions:

- I. The region containing the subject's head and shoulders

II. The region containing the background portion of the image

Each of the methods used for extracting the subject's head from the background employed a scanline-based edge detection mechanism during the edge extraction process. In the instance where a constant background was used, a foreground image containing the subject's head was passed directly to the edge detection mechanism. On the other hand, when an arbitrary background was used, an image subtraction step was required prior to the use of the edge detection mechanism, as the edge detection process was applied to a differential image, not the foreground image. This differential image was obtained by subtracting the background image (obtained during the calibration step) from the foreground image scene containing the subject. The procedure is summarized in equations 10-11. Naturally, the manual feature extraction procedure did not require any sort of subject extraction phase as facial point data were collected manually.

3.4.1 Subject Positioning & Image Capture

During the subject acquisition phase, an individual was placed in a chair, several feet away from the video camera, and an image was captured after the individual had been counselled on how to pose in an optimal manner. Cooperation on the part of the subject was required. In a typical "ideal" pose, the subjects head, neck and shoulders would occupy roughly 40 to 60 percent of the foreground image. No constraints or requirements based on this area were enforced, however, the occluding boundaries of the subjects head needed to be clearly visible and contained within the image. Figure 3.10 illustrates what is meant by an "ideal" view of a subject. Once a suitable image of the subject's face had been obtained it was not subjected to any filtering or pre-processing. Traditional image smoothing techniques, such as median filtering, might have been utilized at this point to reduce the amount of noise present; although levels of background noise were not excessive; much of this was eliminated simply by discarding the least significant image bit-plane. Had this type of operation been required, the 3×3 median filter template shown

in Figure 3.11 would have been applied to the image; undoubtedly at a substantial cost in terms of execution performance.

Figure 3.10 An ideal view of the subject during the image acquisition phase.

3.4.2 The Image Subtraction Step

When arbitrary backgrounds were used, an image subtraction step was necessary before invocation of the edge detection mechanism. By subtracting the background image from the foreground facial image, background components common to both images would cancel each other, thereby enabling the facial region in the foreground to be isolated.

Thus, the foreground image containing the subject's head and shoulders was stored in the image matrix F after having been reduced in the same manner as the background image matrix B (see equation 9). As in the case of the calibration step, the reduction step was carried out purely for performance reasons. The matrix B was subtracted from

the matrix F to produce a resultant image matrix S as outlined in equations 10–11.

$$S_{xy} = |F_{xy} - B_{xy}|, \text{ where } |F_{xy} - B_{xy}| \geq \tau, \text{ for } x = 0, \dots, 127, y = 0, \dots, 479 \quad (10)$$

$$S_{xy} = 0, \text{ where } |F_{xy} - B_{xy}| < \tau, \text{ for } x = 0, \dots, 127, y = 0, \dots, 479 \quad (11)$$

To summarize the process, if elements of F and B differed in absolute value by more than the pre-assigned threshold τ , the resultant value was recorded in the corresponding element of S , otherwise a 0 was stored.

3.4.3 Occluding Edge Detection

Using either a foreground image with a constant background or the image matrix S as input, the left and right¹² occluding edges of the images were obtained, and the resultant edge coordinates were recorded in a set of occluding edge vectors. Traditional edge detection masks and operators such as those illustrated in Figure 3.11 were not used to locate the vertical and horizontal edges in the image. Initially, application of conventional edge detection mechanisms appeared intuitively appealing because the area of edge detection has an appreciative theoretical basis and is well founded in the existing literature. Two popular edge detection operators currently in use are the Laplacian and Sobel operators, which are described in detail by Gonzales [9] and by Pratt in [10].

As shown by Gonzales, if one were to treat an image as a continuous function $f(i, j)$, then the first order partial derivatives or gradients ∇ of the image with respect to x and y can be approximated by

$$\nabla_x f(i, j) = f(i, j) - f(i - 1, j) \approx \frac{\partial f}{\partial x} \quad (12)$$

$$\nabla_y f(i, j) = f(i, j) - f(i, j - 1) \approx \frac{\partial f}{\partial y} \quad (13)$$

¹² Note, that unless otherwise indicated, diagrammatical references to left and right are taken to be from the viewer's perspective.

Figure 3.11 Some commonly used edge detection and filtering templates.

or simply by using absolute differences between horizontally and vertically adjacent elements contained in the image.

$$\begin{aligned}
 z(x, y) &\approx \sqrt{\{\nabla_x f(i, j)^2 + \nabla_y f(i, j)^2\}} \\
 z(x, y) &\approx |\nabla_x f(i, j)| + |\nabla_y f(i, j)| \\
 z(x, y) &\approx \max(\nabla_x f(i, j)^2, \nabla_y f(i, j)^2)
 \end{aligned} \tag{14}$$

A simple approximation involves calculating the magnitude of the digital gradient $z(x, y)$ at a given point in the image or selecting the vertical or horizontal component which is greatest (as shown in 14). The Sobel edge detection operators

$$\begin{aligned}
 \nabla_x f(i, j) &= [f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)] \\
 &\quad - [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)]
 \end{aligned} \tag{15}$$

$$\begin{aligned}
 \nabla_y f(i, j) &= [f(x+1, y-1) + 2f(x, y-1) + f(x+1, y+1)] \\
 &\quad - [f(x-1, y-1) + 2f(x, y-1) + f(x-1, y+1)]
 \end{aligned} \tag{16}$$

are based on this principle. The elements of the two Sobel masks shown in Figure 3.11 correspond to the coefficients for each of the terms of equations 15–16. Gonzales

also shows that the Laplacian operator, based on second order partial derivatives, can be described by

$$\left(\frac{\partial^2 f}{\partial x^2}\right) + \left(\frac{\partial^2 f}{\partial y^2}\right) \quad (17)$$

where the corresponding Laplace expansion is

$$\nabla^2 f(i, j) = f(i + 1, j) + f(i - 1, j) + f(i, j + 1) + f(i, j - 1) - 4f(i, j) \quad (18)$$

This represents the basis for a Laplacian edge detection operator. The elements of the Laplacian mask shown in Figure 3.11 correspond to the coefficients for each of the terms of equation 18.

Approx. Time (Seconds)	Description
3 - 5	Subtraction of two 480 x 128 image, and isolation of occluding edge vectors. (w/ arbitrary background)
8 - 12	Isolation of occluding edge vectors based on a horizontal scanline edge detection mechanism. (w/ constant background)
40	Application of a 3 x 3 Laplacian edge detection mask to the entire 480 x 512 image.
70	Simultaneous application of two 3x3 Sobel edge detection masks to the entire image.

Figure 3.12 Execution times for several edge isolation and detection mechanisms.

Unfortunately, the application of a Laplacian or Sobel edge detection mask such as those shown in Figure 3.11 turned out to be impractical. Both operators were susceptible to local noise or minor irregularities in the image which resulted in the registration of false edges. As well, execution time requirements for a single pass over an image using any of these operators was simply unacceptable. However, had their use been required, the Sobel operators would have been chosen as they appeared to produce substantially

better data than the Laplacian operator. The first two entries of Figure 3.12, correspond to procedures which were actually used in the subject extraction process. The substantial difference in execution time¹³ is indicative of the performance degradation that certainly would have occurred had a conventional edge detection mechanism been used in their place.

Figure 3.13 Search directions for the left and right occluding edge vectors.

A simple but very efficient scanline-based edge detection mechanism was used to locate the left and right occluding boundaries of the subject's head. For each scan-line in the image, the procedure was used to search from the horizontal extremes moving inwards for the first occurrence of a significant gradient on the horizontal image axis. The search directions for each occluding edge are illustrated in Figure 3.13.

¹³ Executed on a machine with an 80286 CPU.

A vector smoothing function based on a local averaging technique was defined as

$$\gamma(v, n) = \left[\frac{\sum_{m=(n-k)}^{(n+k)} v_m}{(2k+1)} \right], \text{ s.t. } \dim(v) \geq (2k+1) \quad (19)$$

where v represented the vector data, n indexed a local interval to be averaged within the vector, and k determined the length of the interval. Although the averaging technique could have been multi-dimensional, it was determined any benefits obtained would have been slight in comparison to the degradation in performance. The constant ($k = 2$) was largely dependent on the amount of background noise present, and lighting conditions in the local environment. As well, k was chosen to be small enough so execution efficiency would be maximized. Each scanline s was then smoothed in the following manner

$$s'_x = \gamma(s, x), \quad \text{for } x = 0 + k, \dots, 511 - k \quad (20)$$

before being passed on to the edge detection mechanism to reduce the possibility of having various irregularities in the image register as false edges.

The function $\lambda(s, \tau, y, i)$, $i \in \{left, right\}$ was used to denote the edge detection mechanism, where s represented the scanline data, τ was a pre-assigned edge detection threshold and y was a vertical index into the image identifying the scanline to be used. As shown in Figure 3.14, the main components of the mechanism were a search interval of length $B (= 10)$, and an index ϵ (shown as e) which tracked intensity gradients within the interval. As in the case of k , the constant B was largely dependent on the amount of background noise and the surrounding lighting conditions. If a “constant” background was used, the consistency of the background surface and the degree to which it actually remained constant would also have to be considered.

Figure 3.14 The scanline based edge detection mechanism.

The conditions and mappings which follow in (21) summarize the behaviour of the mechanism as it travelled from left to right ($i = left$). (Note, the variable β is used to index the current interval.)

$$\begin{aligned}
 [Initial\ State]: \quad & x \leftarrow 0, \quad I_0 \leftarrow s'_0, \quad \epsilon \leftarrow 0, \quad \beta \leftarrow 0 \\
 [Within\ Interval]: \quad & x \leftarrow x + 1, \quad \epsilon \leftarrow \epsilon + (I_\beta - s'_{x+1}) \\
 [End\ of\ Interval]: \quad & I_{\beta+1} \leftarrow I_\beta + \epsilon, \quad \epsilon \leftarrow 0, \quad \beta \leftarrow \beta + 1 \\
 [Terminating\ Condition]: \quad & |\epsilon| \geq \tau \quad (or \quad x = 512)
 \end{aligned} \tag{21}$$

Conversely, the case of the mechanism travelling (in 22) from right to left ($i = right$) is identical, but reversed .

$$\begin{aligned}
 [Initial\ State]: \quad & x \leftarrow 512, \quad I_0 \leftarrow s'_{512}, \quad \epsilon \leftarrow 0, \quad \beta \leftarrow 0 \\
 [Within\ Interval]: \quad & x \leftarrow x - 1, \quad \epsilon \leftarrow \epsilon + (I_\beta - s'_{x-1}) \\
 [End\ of\ Interval]: \quad & I_{\beta+1} \leftarrow I_\beta + \epsilon, \quad \epsilon \leftarrow 0, \quad \beta \leftarrow \beta + 1 \\
 [Terminating\ Condition]: \quad & |\epsilon| \geq \tau \quad (or \quad x = 0)
 \end{aligned} \tag{22}$$

Starting with an an initial intensity of I_0 ($i = left$) or I_{512} ($i = right$), the mechanism tracked intensity gradients along the scanline so if at any point along s'_x it was found that $|\epsilon| \geq \tau$, the current horizontal position x was assumed to correspond to an edge.

If no edge was located within an interval, then the initial intensity for the subsequent interval was set to reflect the current I_p plus ϵ . For example, in Figure 3.14, the points x_{left} and x_{right} represent solution points. Consequently, if an edge was found, the function returned

$$\lambda(s, \tau, y, i) = [x \mid (|\epsilon| \geq \tau)] \quad (23)$$

which represents the current horizontal location on the scanline at the point at which the terminating condition was satisfied.

When a constant background was used, the left and right occluding edge vectors Ω for the subject's head were established by applying

$$\Omega_{y_i} = \lambda(F_y, \tau, y, i), \quad \text{s.t. } \tau > 0, \quad i \in \{left, right\} \quad (24)$$

to each scanline in the foreground image F . This version of the automated extraction algorithm did not require a calibration step, although no automatic method was established for obtaining the edge detection threshold τ . This value had to be found experimentally, and τ was usually in the range 25 to 45, depending on the lighting conditions in the room where the subject images were acquired. The process for the automated extraction procedure (eg. arbitrary background) was similar. Instead, the occluding edge vectors Ω were established by applying

$$\Omega_{y_i} = \lambda(S_y, \tau, y, i), \quad \text{s.t. } \tau = 1, \quad i \in \{left, right\} \quad (25)$$

to each scanline in the image subtraction matrix S . In this case it was necessary that the edge detection threshold value be set to one, because of the means by which the image subtraction procedure was implemented. Essentially, the image subtraction procedure found all differences between the background and foreground images using an automatically determined threshold τ , but did not go so far as to locate the occluding boundaries. Thus, by setting the edge detection threshold equal to 1, this would ensure the first non-zero element of each scanline (in both directions) would be selected as the occluding edge.

To enable the reader to gain a better understanding of the type of data processed by the edge detection mechanism, Figures 3.15 and 3.16 show two test subject images taken against a constant background, each displayed in the form of a three-dimensional intensity mesh. Actual data from which these Figures were generated would have been suitable as input to the scanline-based edge detection mechanism.

Figure 3.15 A three-dimensional intensity mesh of the area including and surrounding the face of *Subject 1*.

Figure 3.16 A three-dimensional intensity mesh of the area including and surrounding the face of *Subject 2*.

3.5 Orientation Analysis

Many problems due to unsuitable orientation were certainly apparent from an early stage in the course of the research. It was discovered that problems directly related to orientation could rarely be treated with any degree of satisfaction. The problem of detecting and correcting for deviations in the three-dimensional orientation of a subject's head remained a dilemma and is discussed in Chapter 5. However, for small corrections in two dimensions,¹⁴ a suitable algorithm was designed and tested, which could handle this problem reasonably well, although it was not incorporated into either of the automated feature extraction algorithms. This was primarily due to the large computational burden it would have placed on the system. Had it been included, it would have been invoked immediately following the subject extraction phase.

The first step of the algorithm required the calculation of a centre vector Ψ as

$$\Psi_y = \left[\frac{\Omega_{y(left)} + \Omega_{y(right)}}{2} \right], \text{ for } y = w(tb), \dots, w(bb) \quad (26)$$

where each element of Ψ represented the horizontal midpoint between left and right edge vector elements in Ω . The boundary functions $w(bb)$ and $w(tb)$ are described in the following section and loosely refer to the location of the bottom of the screen and the location of the top of the subject's head. Elements of Ψ are shown in Figure 3.17.

An ordinary least squares linear regression was used to estimate the slope and intercept for a symmetry equation $y = \alpha + \beta\Psi$ using the data contained in Ψ . Thus, the estimated symmetry equation was $y_i = \hat{\alpha} + \hat{\beta}\Psi_i + \hat{\epsilon}_i$. The estimated error term is shown as $\hat{\epsilon}$, and the slope and intercept estimates $\hat{\beta}, \hat{\alpha}$, were arrived at in the following manner:

$$\hat{\beta} = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}, \quad \hat{\alpha} = \frac{\sum_{i=1}^n y_i - \hat{\beta} \sum_{i=1}^n x_i}{n} = \bar{y} - \hat{\beta} \bar{x} \quad (27)$$

The motivation for estimating the symmetry equation was to provide a rough indication of the subject's orientation in two dimensions. Based on the assumption there existed one

¹⁴ eg. sideways tilting of the head.

Figure 3.17 Calculation and placement of the symmetry equation and centre vector.

or more axis, about which a face was roughly symmetric, the estimated slope coefficient β ideally would have been parallel to the vertical image axis. The angle α° , representing the difference (in degrees) between the vertical axis and the line of symmetry, was first computed. The angle B° was defined as the convergence criterion for the algorithm. Elements of the edge vectors Ω were then adjusted by an amount needed to correct for the angular difference α° . The boundaries for the image were re-calculated and the entire process was repeated until $|\alpha| \leq B$, or a maximum number of iterations had been exceeded. The algorithm was recursive, and typically came up with a suitably corrected pose in less than five iterations with $B \leq 5^\circ$. An overview of the process is provided in Figure 3.18.

Figure 3.18 Overview of the recursive two-dimensional orientation correction algorithm.

3.6 Isolation of The Head and Facial Boundaries

By locating the occluding edges of the head, and storing the corresponding edge location data in Ω , it was possible for the automated feature extraction algorithm to locate, in more precise terms, the region of the image which encompassed the subject's face. To simplify matters, the procedure was decomposed into three steps. The initial step involved the location and initial definition of the outermost boundaries of the subject's head as well as the location of the neck and shoulder boundary. The second step made use of this information to calculate the dimensions and locations for a "head" window as well as an initial "facial" window. The third and final step involved analysis of image sub-scans within this facial window to locate the approximate centre of the face,

after which the facial window and various other boundaries were re-adjusted and their locations finalized.

3.6.1 Definition of The Primary Facial Boundaries and Location of The Neck & Shoulders

All subsequent image analysis was confined to the region of the image bounded by the left and right edge vectors (Ω). The edge vectors were searched for sets of contiguous non-zero left and right edge pairs. If more than one set was established, the set containing the greatest number of valid edge pairs was selected and the remaining sets were discarded. The data in Ω were then smoothed using the local averaging function¹⁵ as shown by:

$$\Omega'_{y_i} = \gamma(\Omega_{y_i}, y), \text{ for } y = w(bb), \dots, w(tb), i \in \{left, right\} \quad (28)$$

The reason for this was to remove any minor anomalies in Ω resulting from the preceding image subtraction process.

The function $w(boundary)$ was used to denote a scalar representing the position of a symbolic *boundary* within an image. The outer bounds of the subject's head were now defined as:

$$w(bb) = y_{(480)} \quad (29)$$

$$w(lb) = \min(\Omega_{(left)}) \quad (30)$$

$$w(rb) = \max(\Omega_{(right)}) \quad (31)$$

$$w(tb) = \min(y \mid (\Omega_{y_{(left)}} \text{ and } \Omega_{y_{(right)}} \neq 0)) \quad (32)$$

$$w(cb) = \frac{\sum_{y=w(bb)}^{w(tb)} \Psi_y}{|\omega(bb) - \omega(tb)|} \quad (33)$$

$$w(mb) = \frac{w(tb) + w(bb)}{2} \quad (34)$$

¹⁵ Defined by equation 19.

The symbols $\{tb, bb, lb, rb, mb, cb\}$ correspond to the top, bottom, left, right, middle, and centre boundaries of the window. In Figures 3.20–3.19, $\{w(lb), w(tb), w(rb)\}$ are shown as boundaries $\{F, Z, H\}$.

Before the final position of the boundary $w(bb)$ could be established, it was necessary to calculate the vertical position of the subject's neck and shoulders. For both the left and right side of the head, this task was accomplished by finding the vertical offset from the top of the head that minimized the angle between $w(cb)$ and a ray from the centre of the head to a point $\{p \mid p \in \Omega\}$. If, for either side of the head, no minimum angle was obtained, or the optimal point was in close proximity to the top of the head then it was assumed the subjects neck and shoulders were not visible and no change was made to $w(bb)$ (eg. $w(bb)' \equiv w(bb)$.)

Figure 3.20 illustrates the process of locating the shoulder-neck boundary. Using the left side of the head as an example, the search for a point $\{p \mid p \in \Omega_{(left)}\}$ that minimizes angle a begins at point A and continues to point G to produce the shoulder-distance vector Υ . By searching this vector, we find the optimal solution is at point C. Similarly, for the right side of the head, I is chosen as the point which minimizes the angle b . The shoulder boundary is then computed as the vertical average of points C and I.

$$\Upsilon_y = \frac{|w(tb) - y|}{|w(cb) - \Omega_y|}, \text{ for } y = w(bb), \dots, w(tb), \quad i \in \{left, right\} \quad (35)$$

$$w(bb)' = \left(\frac{[y | \min(\Upsilon_{y(left)})] + [y | \min(\Upsilon_{y(right)})]}{2} \right), \quad s.t. y \neq w(tb) \quad (36)$$

Equation 35 illustrates how the shoulder-distance vector Υ was computed and equation 36 shows how the boundary $w(bb)' \equiv w(neck)$ was finally re-adjusted to reflect the location of the subject's neck.

Figure 3.19 Location of the initial head boundaries.

Figure 3.20 The search for the neck and shoulder boundary.

3.6.2 Placement of The Facial Window

Based on the location of the previously established head boundaries, a "head" window that encompassed the region of the image containing the subject's head was established as illustrated in the following Figure:

Figure 3.21 Final placement of the head boundaries and head window.

Not only was all subsequent processing and analysis confined to the region of the image bounded by the head window, but its location and dimensions were also used to compute the relative coordinates and dimensions of a somewhat smaller facial window. The reasons for having a facial window were twofold: The first was to reduce the computational complexity of the problem by narrowing down the overall search area, while the second was to bound the area of the image containing important fiducial features

used for identification. The dimensions and initial placement of this facial window were defined in the following manner¹⁶:

$$w(lb_f) = w(cb) - \frac{|w(lb) - w(rb)|}{4} \quad (37)$$

$$w(rb_f) = w(cb) + \frac{|w(lb) - w(rb)|}{4} \quad (38)$$

$$w(tb_f) = w(mb) - \frac{|w(tb) - w(bb)|}{3} \quad (39)$$

$$w(bb_f) = w(mb) + \frac{|w(tb) - w(bb)|}{3} \quad (40)$$

The initial facial window dimensions (relative to the size of the head window) were determined heuristically. The vertical dimension was chosen to slightly exceed the maximum expected distance between the eyes and the mouth, whereas the horizontal dimension was chosen to be slightly greater than the maximum expected distance between the pupils of the eyes. Once an initial position for the facial window had been decided, it was then relocated so that it was vertically centered over the subject's face.

$$s_y = \frac{\sum_{x=\Omega_{y(left)}}^{\Omega_{y(right)}} I_{x,y}}{|\Omega_{y(left)} - \Omega_{y(right)}|}, \quad \text{for } y = w(mb_f), \dots, w(bb_f) \quad (41)$$

As outlined in equation 41, the mean value for each facial sub-scan between the occluding edge vectors, over the vertical sub-range of the facial window were placed in a vector s . The search for the mean brightest s was initiated downwards from $w(mb_f)$ to reduce or eliminate hair as a factor in the determination of $w(mb)'$. The middle boundary $w(mb)$ was adjusted so its vertical position coincided with the mean brightest sub-scan in s .

$$w(mb)' = [y | \max(s_y)], \quad \text{for } y = w(mb_f), \dots, w(bb_f) \quad (42)$$

The facial window was then moved along the vertical axis until it was centered about $w(mb)'$, as shown in Figure 3.22.

¹⁶ The subscript f serves as a reminder that boundaries belonging to the facial window, not the head window are being described.

Figure 3.22 Placement and adjustment of the facial window.

3.7 Locating The Subject's Eyes

The first of the fiducial features to be located were the subject's eyes. The reason for this was twofold: first, the pupils of the eyes could be consistently located with a high degree of precision, and second, the relative eye coordinates were required in defining search constraints for the other features. The algorithm required the eyes be separated by $w(cb)$ as the search for the left and right eye was conducted independently. Two "eye" windows were first established based on the coordinates and dimensions of the facial window :

$$\begin{aligned}
 w(tb_e) &= w(tb_f) \\
 w(bb_e) &= w(mb) \\
 w(rb_{e(t,f)}) &= w(cb) \\
 w(lb_{e(t,f)}) &= w(lb_f) \\
 w(rh_{e(r,q,h,t)}) &= w(rh_f) \\
 w(lb_{e(r,q,h,t)}) &= w(cb)
 \end{aligned} \tag{43}$$

The region of the image encompassed by both eye windows¹⁷ corresponded exactly to the top-half region of the facial window as shown in Figure 3.23

Figure 3.23 Placement of the left and right eye windows.

Equation 44 describes a screen normalized distance function which was used to calculate the Euclidian distance between a pair of points

$$\delta(p_1, p_2) = \sqrt{(p_{1x} - p_{2x})^2 + (\varphi(p_{1y} - p_{2y}))^2} \quad (44)$$

where the screen aspect ratio is represented by the constant φ . The function $\rho(x, y)$ was used to denote the screen coordinates for symbols representing cartesian points. For example, coordinates at the origin of the image would have been represented as $\rho(origin) = \rho(0, 0) = \{0, 0\}$.

A pixel clustering algorithm was used to locate each of the subject's eyes. This first required a frequency distribution

$$f_{l_{x,y}} \leftarrow f_{l_{x,y}} + 1, \text{ where } 0 \leq I_{x,y} \leq 255, \text{ for } y = w(tb_e), \dots, w(bb_e), \quad (45)$$

$$x = w(lb_e), \dots, w(rb_e)$$

¹⁷ The subscript *e* indicates a boundary of either eye window, while $e_{(left)}$ refers to the left eye window and $e_{(right)}$ refers to the right eye window. The subscript *f* indicates a boundary of the facial window.

as well as the mean intensity value $\bar{m}$ for both eye windows to be computed as shown in equations 45 and 46.

$$\bar{m} = \left[\frac{\sum_{y=w(tb_e)}^{w(bb_e)} \sum_{x=w(lb_e)}^{w(rb_e)} I_{x,y}}{|w(tb_e) - w(bb_e)| \times |w(lb_e) - w(rb_e)|} \right] \quad (46)$$

For each eye window, all pixels greater than $\bar{m}$ were immediately discarded, while the darkest η ($\approx 0.05\%$) of the remaining pixels were retained for processing by applying a thresholding procedure. A suitable value for η was determined experimentally. The actual threshold τ used here was calculated as:

$$\tau = \min \{ i \mid f_i \geq \eta \}, \text{ where } 0 \leq i \leq 255. \quad (47)$$

The following procedures were performed on each eye window: Selected pixels were separated into clusters; a cluster being recursively defined as a pixel having one or more neighbors. Unit clusters containing a single point were ignored. Once all clusters had been identified, the centroid of each was computed, as well as its density and size. The centroid or mean coordinates for cluster i were determined in the following manner:

$$\rho(\text{centroid}_i) = \{ \mu_{i(x)}, \mu_{i(y)} \} = \left\{ \frac{\sum_{z=1}^n c_{i(x)} }{n}, \frac{\sum_{z=1}^n c_{i(y)} }{n} \right\}. \quad (48)$$

Based on this information, the coordinates for a base or reference point (later required during the ranking of individual clusters) were calculated as

$$\nu = \frac{\sum_{z=1}^n c_{i(y)}}{n} \quad (49)$$

$$\rho(\text{base}) = \{ w(cb), \nu \} \quad (50)$$

where ν represented the average vertical position of for each of the n clusters in the eye window. The horizontal position of the base point was chosen as $w(cb)$ instead of an

anticipated point for the eye pupil, because of the variability in the width of the head ($\Omega_{v(e,gh)} - \Omega_{v(l,f)} - \Omega_{v(r,f)}$) at point v . In turn, the distance vector α was computed as

$$\alpha_i = \sigma(\mu_{i(x)}, \mu_{i(y)}, \rho(\text{base})), \text{ for } i = 1, \dots, n \quad (51)$$

with α_i representing the distance from the base point to the centroid of cluster i .

The number of pixels belonging to each cluster (eg. the cluster size) was denoted by s_i . For each eye, the n clusters were ranked (equation 52) as a function of their distance from the base point and by their size

$$z = \left[i \mid \max \left(\frac{(1 - \alpha_i) s_i}{\max(\alpha) \sum_{j=1}^n s_j} \right) \right], \text{ for } i = 1, \dots, n \quad (52)$$

so that $c_{(eye)} = c_z$. The centroid of the highest ranked cluster was then selected as $\rho(c_{eye}) = \{\mu_{z(x)}, \mu_{z(y)}\}$, which represented the selected coordinates for the eye. In equation 52, the term $(1 - \alpha_i) s_i$ ranked the clusters, while $\max(\alpha) \sum_{j=1}^n s_j$ was a normalization factor designed to keep these values in the range $0.0 \leq \text{value} < 1.0$.

This clustering process was applied independently to both eye windows to obtain the coordinates for the pupils of the left and right eyes. Immediately following, a horizontal to vertical distance ratio between the eye coordinates was computed. If the ratio exceeded a certain tolerance or threshold $\tau (\approx 7 : 1)$, the image was rejected.

Points A through E in Figure 3.24 indicate clusters, and point Z represents the base point. In this particular example, the left eye window is shown, and the centroid of cluster D would have been chosen as the coordinates of the pupil of the left eye. To provide an indication as to the characteristics of real image data surrounding an eye, two sets of scanline¹⁸ plots as well as two three-dimensional intensity maps for test *Subject1* and *Subject2* are provided in Figures 3.25–3.28.

¹⁸ Figure 3.30 indicates the orientation of scanline sets described in this chapter.

Figure 3.24 The pixel clustering algorithm used to locate the pupil of a subject's eye.

Figure 3.25 Horizontal scans taken from the area surrounding the eyes of *Subject1*.

Figure 3.26 Horizontal scans taken from the area surrounding the eyes of *Subject2*.

Figure 3.27 A Three-dimensional intensity mesh of the area surrounding the right eye of *Subject 1*.

Figure 3.28 A three-dimensional intensity mesh of the area surrounding the right eye of *Subject 2*.

3.8 Locating the Nose and Mouth

The coordinates for each eye were used as reference points in obtaining approximate locations for both the nose and mouth. Based on $\rho(eye_{(left)})$ and $p(eye_{(right)})$, two new windows were defined; one for the nose and another for the mouth. The distance between the pupils of the eyes, η , was defined as shown in equation 53 and used to determine the vertical dimensions of each window.

$$\eta = d(\rho(eye_{(left)}), \rho(eye_{(right)})) \quad (53)$$

Thus for the nose and mouth windows the vertical dimensions were determined as follows:

$$w(tb_n) = \frac{w(eye_{(y,left)}) + w(eye_{(y,right)})}{2} \quad (54)$$

$$w(bb_m) = w(tb_n) + \left[2\eta - \frac{\eta}{4}\right] \quad (55)$$

whereas the two shared vertical boundaries were defined as

$$w(tb_m) = w(bb_n) = \frac{w(tb_n) + w(bb_m)}{2} \quad (56)$$

which represented the midpoint between the top of the nose window and the bottom of the mouth window. The combined vertical dimensions of both windows were defined in terms of the distance between the eyes, and needed to be determined experimentally. As searches for the individual facial features were conducted independently, it was important that separating boundaries between features be placed with consistency and accuracy. Hence, definition of the vertical dimension as $\left[2\eta - \frac{\eta}{4}\right]$ almost invariably resulted in the boundary $w(tb_m) = w(bb_n)$ being situated between the bottom of the nose and the upper lip.

The horizontal boundaries for both windows¹⁹ were defined by the horizontal position of the left and right eye,

$$w(lb_m) = w(lb_n) = w(eye_{(x,left)}) \quad (57)$$

$$w(rb_m) = w(rb_n) = w(eye_{(x,right)}) \quad (58)$$

¹⁹ The subscripts *n* and *m* reference the nose and mouth window respectively.

so the width of both windows was simply the distance between the eyes. The placement of the nose and eye windows are shown in Figure 3.29. Figures 3.33–3.34, and 3.37–3.38 show regions of two test images containing the nose and mouth plotted as a three-dimensional intensity mesh. In each case, the borders of the mesh roughly correspond to the borders of the nose or mouth window.

Figure 3.29 Placement of the nose and mouth windows.

To locate a point on the subject's mouth, the average intensity for each horizontal sub-scan within the mouth window was calculated (in 59) and placed in a vector s .

$$s_y = \frac{\left[\sum_{x=w(lb_m)}^{w(rb_m)} I_{x,y} \right]}{|w(lb_m) - w(rb_m)|}, \quad \text{for } y = w(tb_m), \dots, w(bb_m) \quad (59)$$

As shown in eqn. 60, this vector was searched for an overall minimum which was chosen to represent the vertical position of the subject's mouth.

$$w(mlnc) = [y \mid \min \{s_y\}], \quad \text{for } y = w(tb_m), \dots, w(bb_m) \quad (60)$$

Naturally, the subject's mouth had to be closed for this procedure to work with any degree of accuracy. Sample horizontal scan lines taken from the area surrounding the mouth of two test subjects are provided in Figure 3.35 and 3.36.

A similar procedure was used to provide a approximate point on the subject's nose. The analysis was carried out on vertical sub-scans within the nose window to provide a point on the bridge of the nose in close proximity to the tip. Although the nose window could always be placed accurately, this point on the nose could not be placed with any great degree of consistency and was not used in any of the subsequent analysis. This was largely because multiple light sources often indicated several possible solution points, none of which were necessarily correct. The prospect of utilizing shape from shading techniques to recover shape from a nose is discussed in Chapter 5. Horizontal scanlines²⁰ taken from the area surrounding the nose of *Subject1* and *Subject2* are provided in Figure 3.31 and 3.32. A set of vertical scanlines surrounding the nose and mouth of both test subjects are shown in Figure 3.39 and 3.40.

Figure 3.30 The orientation of scanline sets used in the scanline graphs.

²⁰ Figure 3.30 indicates the orientation of scanline sets described in this Chapter

Figure 3.31 Horizontal scans taken from the area surrounding the nose of *Subject1*.

Figure 3.32 Horizontal scans taken from the area surrounding the nose of *Subject2*.

Figure 3.33 A Three-dimensional intensity mesh of the area surrounding the nose of *Subject1*.

Figure 3.34 A three-dimensional intensity mesh of the area surrounding the nose of *Subject2*.

Figure 3.35 Horizontal scans taken from the area surrounding the mouth of *Subject 1*.

Figure 3.36 Horizontal scans taken from the area surrounding the mouth of *Subject 2*.

Figure 3.37 A Three-dimensional intensity mesh of the area surrounding the mouth of *Subject1*.

Figure 3.38 A three-dimensional intensity mesh of the area surrounding the mouth of *Subject2*

Figure 3.39 Vertical scans taken from the area surrounding the nose of *Subject 11*.

Figure 3.40 Vertical scans taken from the area surrounding the nose of *Subject 12*.

3.9 Manual Point Collection and Feature Projection

The process of manually extracting points from a subject's face was obviously far less complex than the procedures required for the automated procedure. As far as the fiducial point collection process was concerned, a system operator, using a graphics pointing device such as a mouse, would obtain the image coordinates for each of the 11 points shown below in Figure 3.41. The input image used could be read in to the frame buffer from disk without any preparation, or it could be obtained using the video camera; in which case a well posed view of the subject (or photograph of the subject) was used. In either case there were no constraints placed on the type of background used.

Figure 3.41 The eleven facial points located manually (for use with 9 & 4 feature classification algorithms).

Once the raw facial data had been collected, some basic transformations were applied, and the results were added to a file containing a set of facial statistics, also referred to in subsequent sections as **FaceMap** data. For any given individual, the raw image coordinates as well as equivalent data in "facial-normalized" form were stored in the **FaceMap** file.

First, the point M , representing the midpoint between the pupils of the eye, was computed as:

$$\rho(M) = \left\{ \frac{w(eye_{(x,left)}) + w(eye_{(x,right)})}{2}, \frac{w(eye_{(y,left)}) + w(eye_{(y,right)})}{2} \right\} \quad (61)$$

Then, the facial-normalized point corresponding to each feature f_i ²¹ was then derived,

$$\rho(f_i) = \left\{ \frac{f_{i(x)} - w(M_{(x)})}{\eta}, \frac{f_{i(y)} - w(M_{(y)})}{\eta} \right\}, \text{ for } i = 1, \dots, 11 \quad (62)$$

as the difference from point M , and normalized by η , the distance between the subject's eyes. It was found that **FaceMap** data stored in this form could be used for several purposes, including predicting the location of features on the subject's face based only on the location of the eyes. To accomplish this, the upper and lower bounds a, b , as well as the mean μ were computed for each feature f_i as

$$a_{i(x)} = \min(f_{i(x)}) \quad (63)$$

$$a_{i(y)} = \min(f_{i(y)}) \quad (64)$$

$$b_{i(x)} = \max(f_{i(x)}) \quad (65)$$

$$b_{i(y)} = \max(f_{i(y)}) \quad (66)$$

$$\mu_i = \frac{\sum_{j=1}^m f_{i,j}}{m} \quad (67)$$

where m represents the total number of **FaceMap** records. Thus, for the entire population, the mean for each feature was computed as shown in equation 68.

$$\rho(f_{i,\mu_i}) = \{ \eta \mu_{i(x)} + w(M_{(x)}), \eta \mu_{i(y)} + w(M_{(y)}) \}, \text{ for } i = 1, \dots, 11 \quad (68)$$

Additionally, feature "windows" based on the range for each feature was also established in the following manner:

$$\rho(f_{i,a_i}) = \{ \eta a_{i(x)} + w(M_{(x)}), \eta a_{i(y)} + w(M_{(y)}) \}, \text{ for } i = 1, \dots, 11 \quad (69)$$

$$\rho(f_{i,b_i}) = \{ \eta b_{i(x)} + w(M_{(x)}), \eta b_{i(y)} + w(M_{(y)}) \}, \text{ for } i = 1, \dots, 11 \quad (70)$$

²¹ The notation $f_{i(x)}, f_{i(y)}$ refers to the x and y value of feature i

Figure 3.42 Feature windows and points used in the projection of FaceMap data.

Hence, if the coordinates for the subject's eyes could be established, an entire set of predicted facial features and boundaries could be computed. Figure 3.42 illustrates this procedure. Let points (1) and (2) represent the coordinates of the eyes, located either automatically or manually. As the point M and the distance η could be computed, it was possible to predict the location for each feature simply by taking the mean projected coordinate (eg. $\rho(f_{i(m)})$) as the expected location for the i 'th feature. As well, feature windows for the remaining points, (3)-(11), could be predicted using the vectors a and b . More specifically, the top-left and bottom-right corner points for each feature window could be represented by $\rho(f_{t_{(a)}})$ and $\rho(f_{b_{(b)}})$ respectively. The primary advantage in using feature projection techniques is that the search area for individual features can be greatly reduced in a very short period of time, resulting in substantially enhanced performance of the system. The other advantage is that projected feature windows can be used to independently verify results from the conventional feature extraction process, leading to enhanced reliability of the system. Figure 3.43 shows the projection of a set

of **FaceMap** data containing 16 faces. The location of facial points (1)-(11) correspond to those shown previously in Figure 3.42.

Figure 3.43 The projection of manually collected **FaceMap** data.
(used by the feature projection algorithm and 4 and 9 feature classification procedures)

Chapter 4

FEATURE CLASSIFICATION AND RECOGNITION

4.1 Facial Measurements and Ratios

The purpose of the algorithms and procedures described in the previous Chapter was to isolate and abstract various points from a facial image. Once these points had been successfully located, they then represented the fiducial basis sets for all subsequent classification and recognition computations.

In all cases, distances between various points in each basis set were used to represent fiducial features. The screen normalized Euclidian distance function (defined as eqn. 44 from the previous Chapter)

$$\delta(p_1, p_2) = \sqrt{\{(p_{1x} - p_{2x})^2 + (\varphi(p_{1y} - p_{2y}))^2\}}$$

was used to compute the distance between fiducial basis points (p_1, p_2) . Again, the constant $\varphi = 1.2$ represents the screen aspect ratio.

Based on empirical observations, the points representing the pupils of the left and right eye, $\{p(eye_{left}), p(eye_{right})\}$, could be consistently located with a high degree of precision. Furthermore, these fiducial points were common to all basis sets. Consequently, it was decided to normalize all other distance computations to the

distance between the eyes (eg. $\delta(\rho(\text{eye}_{(left)}) , \rho(\text{eye}_{(right)}))$), so the effectiveness of the classification and recognition mechanism would be minimally affected by the physical distance between the head of the subject and the video camera lens. Recognition accuracy seemed to degrade slightly when the subject was very far away²² from the camera. This was apparently due to the decrease in spatial resolution of the facial region in the image.

In all cases, the fiducial distances chosen to represent features were selected according to the following criterion where possible:

- I. Minimal change in distance due to changes in facial expression
- II. Minimal changes in distance over a lengthy elapsed time period (eg. time-invariant)
- III. Minimal changes in distance due to two and three-dimensional changes in orientation

Although the above criterion were strongly desired, it was rare for all or even any of these conditions to be thoroughly satisfied. In each case, the final set of features chosen were selected heuristically and could not be rigorously tested by any precise means. In some cases, this selection process was abandoned altogether, because of the small number of features or fiducial points available.

4.1.1 Automated Feature Extraction

An early version of the automated feature extraction procedure was able to approximately locate the left and right corners of the subject's mouth (shown as distance **b** in Figure 4.47). However, this distance was found to be sensitive to changes in facial expression and the automated extraction algorithm had difficulty in locating these points with consistency.

For the final implementation, the distance **b** (in Figure 4.48) was chosen to represent the distance between the midpoint between the eyes and the top of the head. Although **b** was not directly affected by changes in facial expression, it was particularly sensitive to short and long-term changes in hair style and did not represent a significant improvement

²² eg. 10 or more feet from the camera

over the distance b used in the earlier implementation. Normalized distances d between various pairs a, b of fiducial points and boundaries were calculated as

$$d_i = \frac{\eta}{b(\rho(a_i), \rho(b_i))}, \quad a_i, b_i \notin \{eye(left), eye(right)\} \quad (72)$$

where the normalization factor η represents the distance between the subject's eyes. A description of the normalized distance ratios used by the automated feature extraction procedure follow in Figure 4.44. The ratios shown here correspond to those in Figure 4.48.

Feature #	Feature Description	Ratio (in figure)
1	Vertical distance from eyeline to mouth line	z/a
2	Vertical distance from eye line to top of head	z/b
3	Width of face at the mouth level	z/c

Figure 4.44 A description of features composed of distance ratios used by the automated feature classification algorithm to obtain 3 features.

The data available to the algorithm were elements of the occluding edge vectors, the boundary points, and the three fiducial points in the basis set.

4.1.2 Manual Feature Extraction

As facial points were collected manually, no edge vector or boundary data were available to the manual feature extraction procedure. Distances were derived entirely from points in the fiducial basis set²³. In the first instance of the manual extraction procedure, 9 fiducial distances were computed. The point M was used as a base or reference point, where M was computed as

$$\rho(M) = \left\{ \frac{w(eye(x, left)) + w(eye(x, right))}{2}, \frac{w(eye(y, left)) + w(eye(y, right))}{2} \right\} \quad (73)$$

²³ In each case the basis set contained 11 points

representing the geometric midpoint between the subject's eyes. With the exception of the two eye pupil coordinates, all fiducial distances were measured from the point M. Hence, fiducial distances were computed as

$$d_i = \frac{\eta}{\delta(\rho(M), \rho(f_i))}, \quad f_i \notin \{eye(left), eye(right)\} \quad (74)$$

where f represents the 9 remaining points in the basis set, δ represents the distances from M to f and η represents the distance between the subject's eyes. The motivation here was to use *all* available points in the basis set during the feature classification stage. A description of the 9 fiducial distances used are provided in Figure 4.45. (The ratios correspond to those in Figure 4.49.)

Feature #	Feature Description	Ratio (in figure)
1	Distance from M to left eye, left corner	r/a
2	Distance from M to left eye, right corner	r/b
3	Distance from M to right eye, left corner	r/c
4	Distance from M to right eye, right corner	r/d
5	Distance from M to left side of nose	r/e
6	Distance from M to tip of nose	r/f
7	Distance from M to right side of nose	r/g
8	Distance from M to left corner of mouth	r/h
9	Distance from M to right corner of mouth	r/i

Figure 4.45 A description of features composed of distance ratios used by the manual feature classification algorithm to obtain 9 features..

In the second instance of the feature extraction procedure, a total of 4 features were used. The distances chosen were expected to be particularly insensitive to changes in the

vertical three-dimensional orientation of the subject's head, so only horizontal distances were used. This was done so that during subsequent testing, it would be possible to demonstrate negative performance effects resulting from improper orientation. The number of distances used here were similar to the number used by the automated feature classification algorithm, primarily for testing and comparison purposes. These fiducial distances d were computed as follows:

$$\begin{aligned}
 d_1 &= \frac{\eta}{\delta(\rho(\text{nose}_{(left)}), \rho(\text{nose}_{(right)}))} \\
 d_2 &= \frac{\eta}{\delta(\rho(\text{mouthcorner}_{(left)}), \rho(\text{mouthcorner}_{(right)}))} \\
 d_3 &= \frac{\eta}{\delta(\rho(\text{eyecorner}_{(left, left)}), \rho(\text{eyecorner}_{(left, right)}))} \\
 d_4 &= \frac{\eta}{\delta(\rho(\text{eyecorner}_{(right, left)}), \rho(\text{eyecorner}_{(right, right)}))}
 \end{aligned} \tag{75}$$

where the normalization factor η represents the distance between the subject's eyes. These distances $d_1 \dots d_4$ are shown in Figure 4.50 and described in the following Figure. (The ratios correspond to those in Figure 4.50.)

Feature #	Feature Description	Ratio (in figure)
1	Width of the nose	z/a
2	Width of the mouth	z/b
3	Width of the left eye	z/c
4	Width of the right eye	z/d

Figure 4.46 A description of features composed of distance ratios used by the manual feature classification algorithm to obtain 4 features.

Figure 4.47 Distance ratios used by the earliest version of the automated feature classification algorithm to obtain 3 features.

Figure 4.48 Distance ratios used by the final version of the automated feature classification algorithm to obtain 3 features.

Figure 4.49 Distance ratios used by the manual feature classification algorithm to obtain 9 features.

Figure 4.50 Distance ratios used by the manual feature classification algorithm to obtain 4 features.

4.2 Encoding and Storage of Feature Vectors

The fiducial distance measurements described in the previous section were used for facial classification and identification purposes. Once these distances had been computed for an individual, they were encoded in the form of a feature vector, which could then be added to a "feature" file containing feature vectors and other relevant information for individuals known to the system for classification or training purposes. As well, unknown feature vectors could be compared with the set of feature vectors corresponding to individuals known to the system, for identification purposes.

Each feature required 8 bytes²⁴ of storage even though it became apparent this degree of accuracy was not necessary; 4 bytes²⁵ would have been sufficient. To encode an entire feature vector required 24 bytes for the automated 3-feature algorithm, 32 bytes for the manual 4-feature algorithm, and 72 bytes for the manual 9-feature algorithm. Other secondary items of information which were stored with each feature vector in the feature file were as follows:

- I. A record identification number
- II. The subject's name
- III. Image acquisition date
- IV. Comment field

The record identification number served two purposes: first as a unique identification number for each individual, and secondly as a logical record number within the actual feature file. Obviously the subject's name (or some similar form of identification) was required. As well, the classification date was usually included, and the comment field was used to record any peculiarities or unusual information concerning the subject or the way in which the subject had been classified. However, only the record identification number and name fields were absolutely necessary. For each record in the feature file, the additional information overhead required 124 bytes of storage.

²⁴ eg. double precision floating point format

²⁵ eg. ordinary floating point format

For the automated facial classification and recognition algorithm, a plot of the raw feature data for the 28 known subjects is shown in Figure 4.51. In some cases, entire feature vectors for various individuals can be distinguished. The frequency distributions for each feature are given in Figures 4.52—4.53.

Similarly, for the manual facial classification and recognition algorithms, a plot of the raw feature data for the 16 known subjects is shown in Figure 4.54 (using 9 features) and Figure 4.59 (4 features). Although feature vectors for various individuals can not be easily distinguished when 9 features are used, the general characteristics for the set of feature vectors is quite well defined. This was true to a lesser extent when 4 features were used. The frequency distributions for individual features are provided in Figures 4.55–4.58 (9 features) and in Figures 4.60–4.61 (4 features).

Figure 4.51 Automated feature extraction using 3 features: raw feature data.

Figure 4.52 Automated feature extraction using 3 features: frequency distributions for the 1'st and 2'nd features

Figure 4.53 Automated feature extraction using 3 features: frequency distribution for the 3'rd feature.

Figure 4.54 Manual feature extraction using 9 features: raw feature data.

Figure 4.55 Manual feature extraction using 9 features: frequency distributions for the 1'st and 2'nd features.

Figure 4.56 Manual feature extraction using 9 features: frequency distributions for the 3'rd and 4'th features.

Figure 4.57 Manual feature extraction using 9 features: frequency distributions for the 5'th and 6'th features.

Figure 4.58 Manual feature extraction using 9 features: frequency distributions for the 7'th, 8'th and 9'th features.

Figure 4.59 Manual feature extraction using 4 features: raw feature data.

Figure 4.60 Manual feature extraction using 4 features: frequency distributions for the 1'st and 2'nd features.

Figure 4.61 Manual feature extraction using 4 features: frequency distributions for the 3rd and 4th features.

4.3 Classification & Recognition Methodology

After pinpointing various features on the subject's face, fiducial distance ratios between these features were computed and encoded in the form of a feature vector. This feature vector was subsequently added to a "feature file." At a future point in time, the same subject was presented to the system and an unknown feature vector was obtained for testing purposes. A set of Euclidian distance measurements were computed for the unknown feature vector against those stored in the feature file. Measurements were normalized using the computed standard deviation for each feature. Subject records in the feature file were then sorted in increasing order, in terms of their normalized Euclidian distance from the unknown feature vector. The subject was identified as the individual described by the first record in a sorted list of possible candidates.

The classification and recognition methodology used followed a close subset of the techniques employed by Harmon, Goldstein *et al* in [6], [11], [3] and [1]. Particularly in [1], Harmon *et al* were able to demonstrate that high recognition rates could be obtained with a population of relatively large size²⁶, using between 11 and 17 features. These techniques were particularly appropriate for use with this type of application, as all fiducial features were represented by geometric Euclidian distances. Additionally, the techniques were straightforward, simple and efficient, making them particularly well suited for real-time implementation and-or parallel processing. Since feature classification for an individual was based on a single training image, there was no averaging of feature vectors belonging to an individual. The notation used here is similar to that used by Harmon *et al*. (discussed in Chapter 2.) The variables i and j index the feature and individual respectively, and $n \in \{3, 4, 9\}$ represents the number of features used and $m \in \{16, 28\}$ represents the number of individuals known to the system. The set of feature vectors for known persons is denoted by R . The mean μ for each feature was

²⁶ In Harmon's case, 121 subjects were used.

calculated as

$$\mu_i = \frac{\left(\sum_{j=1}^m r_{i,j} \right) + p_i}{m+1}, \quad \text{for } i = 1, \dots, n \quad (76)$$

and the standard deviation s for each feature was computed as

$$s_i = \sqrt{\frac{\sum_{j=1}^m (r_{i,j} - u_i)^2 + (p_i - u_i)^2}{m}} \quad (77)$$

where p represents an unknown feature vector presented to the system for testing purposes.

The Euclidian distance between the unknown feature vector and those known to the system in R was calculated in the following manner.

$$\xi(p, R) = \sqrt{\left\{ \sum_{i=1}^n \left[\frac{p_i - r_{i,j}}{s_i} \right]^2 \right\}}, \quad \text{where } n \in \{3, 4, 9\} \quad (78)$$

A vector, Θ , of pairwise Euclidian distances between the unknown feature vector p and R was defined as

$$\Theta_j = \xi(p, R) \quad (79)$$

The feature vector in R chosen to correspond to the unknown feature vector p was selected as $\min(\Theta)$.

Elements of Θ were then sorted in ascending order so that Θ_1 represented the system's best guess, and also so the remaining elements of R could be ranked in decreasing order of similarity from the unknown feature vector p . Note, however, the semantics or meaning of "similar" and "dissimilar" are quite different in this computational context. For example, similarity between a pair of feature vectors did not mean the corresponding individuals would necessarily look alike.

As the number of known subjects was quite small²⁷, feature windowing techniques²⁸ such as those described in Chapter 2 were not incorporated into the recognition mechanism. Had a substantially larger number of subjects been used, these techniques might

²⁷ The number of subjects known to the system never exceeded 30. Four features were used whereas Harmon used eleven. Cannon *et al* used seventy-five subjects.

²⁸ eg. those used by Harmon *et al* in [1].

have proved beneficial. Furthermore, with such a small population base, execution performance of the recognition mechanisms was not an of immediate concern; the recognition process was virtually instantaneous. It was not known at what point the use of windowing techniques would have been beneficial, although from Harmon's work in [1], it would appear that multiple classified feature vectors for each individual would be required for effective use of this technique.

To gain a better understanding of the Euclidian distance data, each r was compared with a zero feature vector (eg. $p = (0_1, 0_2, \dots, 0_n)$), where n denotes the number of features used. This provided an indication as to how the features were separated, in terms of their distance from the origin of the n —dimensional Euclidian feature space. These Euclidian distance scores and corresponding frequency distributions for each of the 3,9 and 4 feature implementations are shown in Figures 4.62, 4.63 and 4.64.²⁹ The separation between distance scores demonstrate clear and distinguishable differences between individuals do in fact exist; specifically differences which can be used for identification purposes. The C source listing for the primary recognition function is provided in Appendix C for those who find this aspect of the system particularly interesting (as well as for those who wish to see the implementation details).

²⁹ In each of the Euclidian distance score digrams, the vertical axis is meaningless, and is not labelled.

Figure 4.62 Automated feature extraction using 3 features and 28 feature vectors: Raw Euclidian distance scores (top) and corresponding frequency distribution (bottom).

Figure 4.63 Manual feature extraction using 9 features and 16 feature vectors: Raw Euclidian distance scores (top) and corresponding frequency distribution (bottom).

Figure 4.65 Overview of the automated facial classification and recognition algorithm.

Figure 4.64 Manual feature extraction using 4 features and 16 feature vectors: Raw Euclidian distance scores (top) and corresponding frequency distribution (bottom).

4.4 Summary of The Facial Classification & Identification Process

Figures 4.65–4.66 illustrate the arrangement and integration of procedures described in previous sections to form the automated version of the facial classification and recognition algorithm. A similar diagram (Figure 4.67) is also provided for the identification algorithm based on manual feature extraction. Arrows composed of dotted lines show options or procedures which were not directly incorporated into the algorithm; although they could have been with little difficulty.

4.4.1 Automated Feature Extraction & Identification

Facial images were either loaded from disk or acquired directly using a video camera. Live images were acquired against a constant background or an arbitrary one, in which case a calibration phase was required beforehand. Either a scanline-based edge detection mechanism or image subtraction algorithm was used to isolate the region of the image containing the subject's head, and vector data describing the occluding edges of the subject's head were obtained.

The edge vectors were subsequently smoothed using a local averaging function, primarily to remove any irregularities resulting from the image subtraction and edge detection process. Following this step, the edge vectors could have been used as input to an existing two-dimensional orientation correction algorithm. This procedure was circumvented because of the substantial amount of processing and execution time requirements needed to carry out the operation effectively. Instead, the occluding edge vectors were passed to a procedure that located the neck and shoulder boundaries, and if any discontinuities were found in the edge vectors, the image was rejected. The remaining 3 outer boundaries of the subject's head were then established and validated. With the exception of the neck and shoulder boundary, any other boundary that was found to lie

on an outer extremity of the frame was considered invalid, resulting in the rejection of the image.

Upon validation of the head boundaries, the coordinates for a head window were established. Based on analysis of the region enclosed by the head window, several facial boundaries were established and the coordinates for a facial window were defined. If any portion of the facial window was found to lie outside of the region enclosed by the head window, the image was rejected. Search windows for the left and right eyes were then created and the location of initial base reference points for each eye were then computed.

A pixel clustering algorithm was used to establish the coordinates for the pupil of each eye, and these coordinates were then checked to ensure the subject's head was oriented in an acceptable manner. If orientation problems were detected at this stage then the image was rejected. Based on the relative location of the subject's eyes, windows for the subject's nose and mouth were established. The region of the image contained by the mouth window was then analyzed to produce the vertical coordinate of the subject's mouth. This coordinate might have been verified by checking against facial data statistics contained in the **FaceMap**³⁰ point set, however it was found that valid facial images were occasionally rejected when this check was in place.

Distance ratios between various points on the face were computed and subsequently encoded in the form of a feature vector. For classification purposes, the feature vector, along with information describing the person (eg. name, date, comments etc.), was then added to a feature file containing feature vectors and related information for all individuals known to the system. Alternatively, for identification purposes, the feature vector was simply treated as unknown and submitted to the recognition procedure, where it was compared to all known feature vectors in the system. The recognition procedure then produced a list of individuals ranked in order of similarity to the unknown feature vector, based on the 3-dimensional normalized Euclidian distance between the unknown vector and all those known to the system.

³⁰ described in the previous Chapter.

4.4.2 Manual Feature Extraction & Automated Identification

The integration of procedures used by the manual classification and training algorithm are shown in Figure 4.67. For obvious reasons, the entire process was substantially less complex than its automated counterpart. A facial image was either loaded from disk or acquired directly using a video camera. No calibration step was required, nor was *a priori* knowledge of the background specifications or lighting. An operator manually selected 11 predefined facial points using a mouse, and verified the coordinates for each point had been chosen correctly. Upon verification of the raw facial point data, the vertical position of both the left and right sides of the nose were set to the vertical position of the nose tip. The vertical and horizontal distance of each point from the midpoint between the eyes was computed and normalized to the distance between the eyes. The resulting facial data could either be encoded in the form of a feature vector, which was then used for identification purposes, or the data was stored in a file containing facial statistics (eg. the **FaceMap**³¹ point set) used primarily for feature projection purposes. Additionally, the set of feature vectors describing individuals known to the system was generated indirectly from the **FaceMap** data. The recognition procedure was identical to the one used by the fully automated version of the algorithm, except that 4 or 9 features were used.

³¹ described in the previous Chapter.

Figure 4.65 Overview of the automated facial classification and recognition algorithm.

Figure 4.66 Overview of automated facial classification and recognition algorithm (continued).

Figure 4.67 Overview of manual training and recognition algorithm.

Chapter 5

DISCUSSION

5.1 System Performance

5.1.1 Overview

Detailed test results for the individuals, *Subject1* and *Subject2*, are described. Both subjects were tested using the automated 3-feature algorithm, as well as the partially automated 4 and 9 feature algorithms. The data and output for the automated feature extraction and identification algorithm give a clear indication as to the system performance for several “ideal” test cases as well as a “worst-case” test scenario resulting from deliberately caused orientation problems. It would be convenient to provide an optimistic set of test results, or a general identification rate, however, this would not be indicative of the practical capabilities and drawbacks for both the automated and semi-automated approaches. Testing of the automated system yielded *average* recognition rates in the range from 70-80%, although the results could not be reproduced with great consistency. On rare occasions, identification rates would be substantially lower or higher. It is expected these rates would increase for the partially automated identification procedures, primarily because of the greater number of features employed. When misidentification did occur under ideal conditions, the individual was usually ranked as the system’s second or third choice. Misidentification under such conditions should not be seen as a complete

failure; in practical circumstances it would be unrealistic to expect optimal or perfect identification rates. Close matches were of interest as well. For example, a failure or misidentification in which an individual was ranked 2:28 would indicate the individual was the system's second choice out of a possible collection of 28 known subjects, which could hardly be considered a total failure.

The average identification rate for the automated 3-feature algorithm is comparable to Harmon's earlier 50–86% profile identification results, although the highest rates obtained during the course of this research were somewhat lower than the 96% rates obtained by Harmon (in [1]) and Cannon (in [2]). Two factors which directly resulted in improved results for Harmon were the choice of better features and the use of feature thresholding techniques. In all cases, both Harmon and Cannon used a larger number of features. The representation of features (eg. by intensity values) in Cannon's work was entirely different than the geometric representations used here and by Harmon. Since Cannon's identification process was primarily based on intensity values around the eyes, it was highly sensitive or dependent on the local lighting constraints. Lighting arrangements needed to be carefully controlled. Harmon also had several lighting constraints in place when his profile photographs were taken, as high contrast silhouettes were required. However, no specific lighting constraints were in place for this study, except that the subject acquisition room be well lit.

External clues such as the fiducial background mark used by Cannon were not used here either. As well, Harmon and Cannon used between 3 and 8 training images per subject, whereas identification in this study was based on a single training image. Both used feature thresholding techniques to reduce the number of possible candidates during the identification phase, whereas these methods were not utilized by the recognition mechanism described earlier. Harmon started with a large number of possible candidate features and selected only the best features after lengthy analysis. Unfortunately, only 3 features were available to the automated identification program, so analysis and reduction of the number features used would have been futile. Since distance ratios were used, 2 of the 3 features were *absolutely* necessary. Since Harmon was using a facial profile,

the task of locating an abundance of fiducial points was relatively simple — this led to the use of a large number of features and resulted in very high identification rates. Conversely, the underlying strength in Cannon's approach was that he was able to make full use of his carefully constrained environment to obtain similar identification rates. Identification rates for the automated identification procedure described here were lower for primarily two reasons. First, full facial images were used instead of profiles, which meant a significantly smaller number of features could be located and used, and secondly, every attempt was made to minimize lighting and other local processing constraints, so other types of feature information (eg. based upon regions of facial intensities) could not be used.

5.1.2 Selection of Test Images

A collection of facial images of *Subject1* and *Subject2* were obtained so that the algorithms proposed in this thesis could be tested. For each subject, 15 images were acquired and saved on disk. The first 10 of the 15 were "ideal" well posed images, whereas the remaining 5 were obtained with the three-dimensional orientation deliberately mis-aligned. In the case of the mis-aligned poses, the subject's head was tilted up approximately 15 to 30 degrees. From each set of 15 images, 4 test images were eventually selected using the automated identification algorithm. Both *Subject1* and *Subject2* were known to the system, although as far as the system was concerned, the test images were simply of known individuals whose identity had not yet been established.

First, the two sets of 15 images were processed by the automated identification algorithm, so for each of the 30 images the Euclidian distance data and ranked list of matched candidates was recorded. Then, from the "ideal" set of 10 images two test images were selected that resulted in correct identification. If one or more of the remaining "ideal" images were misidentified, the image with the most favorable rank was selected as a third test image. If none of the 10 "ideal" images had been misidentified, a

third test image corresponding to a correct identification was selected from this set. The fourth test image was always selected as the worst³² from the 5 in the “bad orientation” set. These selection criteria were designed to provide a realistic collection of test images for analysis over a range of different conditions.

5.1.3 Selection of Test Feature Vectors

Two sets³³ of 4 test feature vectors were obtained for each of the identification programs. For the automated program, each set of 3-dimensional test vectors was obtained during the test image selection process. On the other hand, the process of generating a set of 4 or 9-dimensional test vectors for each of the semi-automated programs was slightly more elaborate. Two sets of facial points were obtained manually for each test image and both sets were automatically processed by the recognition procedure to produce two test feature vectors. Later, the test vector which produced the best identification results was retained, and the other was discarded. By doing this, it was assumed the possibility of incorrect identification due to errors in the manual placing of facial points would be minimized.

In the tables and Figures which follow, the fourth test corresponds to the worst-case identification test for both *Subject1* and *Subject2*. As well, for *Subject1* the misidentified image corresponds to the first test, and for *Subject2* the misidentified image corresponds to the third test. The final set of tested feature vectors are presented with their corresponding sets of classified (known) feature vectors in Figures 5.68–5.70. For the fully automated implementation (Figure 5.68), feature vectors **V1** and **V2** represent *Subject1* and *Subject2*, whereas in both partially automated versions (Figures 5.69, 5.70), feature vectors **V3** and **V16** correspond to *Subject2* and *Subject1*.

³² eg. the most incorrectly identified facial image

³³ eg. one set for *Subject1* and another for *Subject2*

Figure 5.68 Feature vector data (3 features) used by the fully automated identification algorithm (top), and sets of 4 test feature vectors for *Subject1* and *Subject2*.

Figure 5.69 Feature vector data (9 features) used by the partially automated identification algorithm (top), and sets of 4 test feature vectors for *Subject1* and *Subject2*.

ID	F1	F2	F3	F4
V1	1.54873	1.21557	0.70629	1.98302
V2	1.75911	1.45755	0.68475	1.85110
V3	1.53287	1.42630	0.64578	1.68181
V4	1.58138	1.54648	0.67615	1.81694
V5	1.77044	1.33771	0.70053	1.84191
V6	1.44663	1.22103	0.65263	2.04345
V7	1.56751	1.40781	0.68067	1.88744
V8	1.51950	1.29604	0.68744	1.83092
V9	1.39916	1.17622	0.66667	1.80790
V10	1.76938	1.37127	0.69041	1.80208
V11	1.92332	1.31596	0.72105	1.74170
V12	1.75013	1.35951	0.72526	1.91959
V13	1.50960	1.24468	0.72735	1.77798
V14	1.44584	1.49657	0.69192	1.73882
V15	1.44383	1.37518	0.71024	1.79688
V16	1.48262	1.27258	0.71589	2.09996

Subject1: 4 test feature vectors				
Test#1	1.67412	1.28323	0.68754	2.13441
Test#2	1.60939	1.26189	0.68942	2.15997
Test#3	1.67365	1.24256	0.68915	1.99936
Test#4	1.59114	1.24904	0.70717	2.05402

Subject2: 4 test feature vectors				
Test#1	1.48078	1.17501	0.62716	1.71121
Test#2	1.47232	1.35890	0.64611	1.79635
Test#3	1.44316	1.31308	0.63323	1.76015
Test#4	1.51563	1.37417	0.63201	1.77695

Figure 5.70 Feature vector data (4 features) used by the partially automated identification algorithm (top), and sets of 4 test feature vectors for *Subject1* and *Subject2*.

Figure 5.71 *Subject1*. recognition scores for 4 test feature vectors, ranked against 28 known vectors, using automated extraction of 3 features. (top - Test Score1, 2nd from top - Test Score2, 3rd from top - Test Score3, bottom - Test Score4)

Figure 5.72 Subject1: recognition scores for 4 test feature vectors, ranked against 16 known vectors, using manual extraction of 9 features. (top - Test Score1, 2nd from top - Test Score2, 3rd from top - Test Score3, bottom - Test Score4)

Figure 5.73 *Subject1*: recognition scores for 4 test feature vectors ranked against 16 known vectors, using manual extraction of 4 features. (top - Test Score1, 2nd from top - Test Score2, 3rd from top - Test Score3, bottom - Test Score4)

Figure 5.74 *Subject2*: recognition scores for 4 test feature vectors, ranked against 28 known vectors, using automated extraction of 3 features. (top - Test Score1, 2nd from top - Test Score2, 3rd from top - Test Score3, bottom - Test Score4)

Figure 5.75 *Subject2*: recognition scores for 4 test feature vectors, ranked against 16 known vectors, using manual extraction of 9 features. (top - Test Score1, 2nd from top - Test Score2, 3rd from top - Test Score3, bottom - Test Score4)

Figure 5.76 Subject2: recognition scores for 4 test feature vectors, ranked against 16 known vectors, using manual extraction of 4 features. (top - Test Score1, 2nd from top - Test Score2, 3rd from top - Test Score3, bottom - Test Score4)

Description	# of Feature Vectors or Subjects	Test Image# (for each subject)	Subject1	Subject2
			Rank	Rank
Automated Feature Extraction 3 Features	28	1	2	1
		2	1	1
		3	1	2
		4 (bad orientation)	6	9
Manual Feature Extraction 9 Features	16	1	1	1
		2	1	3
		3	1	1
		4 (bad orientation)	6	4
Manual Feature Extraction 4 Features	16	1	1	2
		2	1	1
		3	4	1
		4 (bad orientation)	2	1

Figure 5.77 A summary of recognition test scores based on 3 sets of 4 test feature vectors (from the 4 test images) for both *Subject1* and *Subject2*.

5.1.4 Identification Test Results

Results for each of the identification tests are summarized in Figure 5.77. Test scores were computed as the pairwise distance $\xi(p, R)$ between a test feature vector p and those known to the system in R . For each score, a rank was assigned to each of the known feature vectors in R , based on the distance from the unknown test feature vector. The vector in R closest to the test vector p was ranked first, while others were ranked in increasing order in terms of their distance from p . Therefore, in Figure 5.77, a rank of 1 corresponds to correct identification (the system's 1'st choice), whereas a rank of n , where ($n > 1$), corresponds to the system's n 'th choice and hence a misidentification.

In the fully automated case, the collection of known subjects consisted of 2 females and 26 males, of which 17 were caucasian, 5 oriental, 4 east-indian, 1 was middle-eastern and 1 was aboriginal. The majority³⁴ of the 28 known subjects were undergraduate students in their late teens to early twenties. The 16 known subjects for both manual feature extraction and automatic identification programs were simply a subset of the 28 subjects used for the automated version. In this case, the population of known individuals contained 2 females and 14 males, of which 8 were caucasian, 4 oriental, 2 east-indian, 1 was middle-eastern and 1 was aboriginal. As well, *Subject1* was caucasian, *Subject2* was oriental, and both subjects were male. At the time of testing, the training image³⁵ for *Subject1* was 13 months old, while the training image for *Subject2* was 2 months old.

For each of the test scores in Figure 5.77, graphs corresponding to the test data are provided in Figures 5.71–5.76. Each graph represents a single test, where the Euclidian distance $\xi(p, R)$ between the the unknown test feature vector p , and the set of known feature vectors R , is plotted against the corresponding rank for each r_i . Thus, the relative Euclidian distance between p and each r_i , as well as the relative distances between adjacent r_i 's provide an indication as to the performance of the identification mechanism. For example, the first and third test scores in Figure 5.72 show the mechanism performed well by choosing *Subject1* from the remaining population of known individuals because in each case, the Euclidian distance between the unknown feature vector p and the first vector r_1 is considerably smaller relative to the Euclidian distance(s) between p and the remaining vectors $r_2 \dots r_m$. With regard to *Subject2*, the same conclusions can be drawn for the first test score in Figure 5.74 and the third score in Figure 5.75. As well, the first test score in Figure 5.73 corresponds to a correct identification, although the relative difference between the subsequent choice is rather marginal; an incorrect identification could have easily resulted. Under "ideal" circumstances, the partially automated 4 and 9 feature implementations managed to produce correct identifications in all cases where the fully automated 3-feature version had failed. However, on several other³⁶ occasions,

³⁴ approximately one third of the subjects

³⁵ eg. the facial image used for feature vector classification.

³⁶ eg. test score 3 in Figure 5.73 and test score 2 in Figure 5.75

they failed where the 3-feature implementation had not. Evidently, sensitivity to small changes in orientation was substantial. The remaining graphs (in Figures 5.71–5.76) correspond to test scores which illustrate typical behaviour of the system.

In most cases where incorrect identification did occur, the relative Euclidian distance between the incorrectly chosen feature vector and the correct one was slight. This is illustrated in Figure 5.71 by the first test score, the third test score in Figure 5.74, and the second test score in Figure 5.75. In all cases where normal or “ideal” images were the source, small deviations in orientation were largely responsible for incorrect identification, as test images for both subjects were acquired during the course of a single sitting.

As expected, the fourth test score in Figures 5.73 and 5.76 indicated the 4-feature implementation was able to overcome the deliberately induced orientation problems which resulted in chaos for the 3 and 9 feature implementations, while performing adequately under normal circumstances.

5.1.5 Processing Speed

On an Intel 80286 based machine, the calibration step and the subject extraction phase of the fully automated algorithm each required approximately 5 seconds of processing time. The subsequent feature isolation procedures usually required between 1 and 6 seconds of processing time, as this would vary directly with the size and dimensions of the fiducial head window. On the other hand, execution of the equivalent process, using a constant background, often required up to 20 seconds of processing time. As the subject population used was small, the actual process of identification³⁷ was virtually instantaneous, although this would certainly change with a significant increase in the number of individuals known to the system.

The author ported a large subset of the code for the automated calibration and extraction procedures to an embedded real-time system based on a single processor

³⁷ eg. retrieval of the feature vector set (F) from disk, and Euclidian distance calculations

(Motorola 68010), and a powerful Matrox MVP-VME image processing board equipped with it's own 68010 processor, running under the Unison operating system. Execution times were at least 10% slower than on the 80286 based machine (with virtually no modification to the original code). However, it is expected if the application code had been tailored specifically to this environment, and if the MVP's firmware had been exploited, execution times would have been remarkable.

5.2 Facial Shape Recovery

Although most commonly employed in the area of robotics, the “shape from shading” paradigm is certainly worthy of investigation. While the scope of this area is broad, the use of a shape from shading model in the context of facial identification was appealing for the following reasons:

- I. Shape information might be used to improve recognition capabilities by providing additional information describing various facial features and structures.
- II. Shape information might also be used for three-dimensional orientation correction.

Investigation of the area was conducted experimentally. The primary aim in this experimentation was to locate the prominent light sources illuminating a subject, by analyzing images of a regular, known object placed in the same position where the human subject would ordinarily be situated.³⁸ As well, a quantized reflectance map for the object was obtained.

5.2.1 Light Source Orientation

To explore the feasibility of implementing a shape recovery mechanism, a simple experiment was designed using a ping-pong ball as a test object. The reasons for using a ping-pong ball were twofold. First, the spherical properties of the ball were particularly simple to model, and second, the plastic surface was relatively non-specular, thereby making it somewhat easier to analyze. Furthermore, orientation problems were known to be non-existent when imaging a spherical object.

First, an image of a ping-pong ball placed in front of a constant background was acquired. An intensity mesh depicting the illumination characteristics of this image is shown in Figure 5.78, in which two prominent light sources are visible. The ball was

³⁸ eg. during the image acquisition phase.

located at approximately the same location where subjects were normally positioned, so light source directions could be established for a particular location in the room. Following this, the occluding left and right edges as well as the top and bottom boundaries of the ball were extracted from the image. The techniques used for edge extraction were similar to those used for facial extraction. Occluding edge vectors Θ , for the ball, were computed as

$$\Theta_{y_i} = \lambda(I_y, \tau, y, i), \quad \text{s.t. } \tau > 0, \quad \text{for } y = 0, \dots, 479. \quad i \in \{left, right\} \quad (80)$$

where the function λ denotes the edge detection mechanism described in Chapter 3. The top boundary was taken to be $\min(y \mid (\Theta_{y(left)} \text{ and } \Theta_{y(right)} \neq 0))$ and the bottom boundary as $\max(y \mid (\Theta_{y(left)} \text{ and } \Theta_{y(right)} \neq 0))$. Similarly, the left boundary was simply $\min(\Theta_{(left)})$ and the right boundary $\max(\Theta_{(right)})$. As in the case of facial extraction (using a constant background), the value assigned to τ was determined by experimentation. The first point to be located was the centre of the ball, as this was later required for determining surface orientation. Based on the outer boundary positions, the centre of the ball was computed as shown in equation 81.

$$p(\text{centre}) = \left\{ \frac{w(lb) + w(rb)}{2}, \frac{w(tb) + w(bb)}{2} \right\} \quad (81)$$

The use of a centre point between the horizontal and vertical boundaries managed to work adequately. To simplify subsequent computations, the centre coordinate of the ball was treated as the logical origin of the image.

Next, the horizontal and vertical diameter d_1, d_2 , of the ball was measured as outlined in equations 82–83 so the radius could be obtained.

$$d_1 = \delta(w(lb), w(\text{centre}_{(y)}), w(rb), w(\text{centre}_{(y)})) \quad (82)$$

$$d_2 = \delta(w(tb), w(\text{centre}_{(x)}), w(bb), w(\text{centre}_{(x)})) \quad (83)$$

The function $\delta(x_1, y_1, x_2, y_2)$ (defined by eqn. 44) was used to compute the screen-normalized distance between each of the two points. As well, a check was made to ensure $\left| \frac{d_1}{d_2} \right| \approx 1$, or very close to it, as any other result would have implied either

the edge-boundary detection algorithm had failed, or the object being analyzed was not circular. Once the boundary values had been validated, d_1 and d_2 were used to calculate the radius r of the ball as shown in equation 84. Placement of the outer boundaries are illustrated in Figure 5.79.

$$r = \frac{\left(\frac{d_1 + d_2}{2}\right)}{2} \quad (84)$$

Since it was known to the algorithm that a spherical surface was being analyzed, it was possible to infer surface orientation (p, q) could be obtained at a given (x, y) coordinate as shown in equations 85–86,

$$p = \frac{x}{\sqrt{|r^2 - (x^2 + y^2)|}} \quad (85)$$

$$q = \frac{y}{\sqrt{|r^2 - (x^2 + y^2)|}} \quad (86)$$

with the search region in the image constrained by $x^2 + y^2 \leq r^2$. The values p and q correspond to the horizontal and vertical gradients at any given point in the image.

To determine the orientation of one or more³⁹ prominent light sources, the algorithm started with four solution points placed at the centre of each visible quadrant of the sphere. From each of these points, a gradient search mechanism was used to independently locate an overall intensity maximum subject to $(x^2 + y^2 \leq r^2)$.⁴⁰ Duplicate solutions for each point (if any) were discarded, and the coordinates for the unique solutions were saved. Thus, if n unique solution points were located, it was assumed n prominent light sources were present. In the case of the ball depicted in Figure 5.78, there are two unique solution points which correspond to points A and B in Figure 5.79.

The assumption was made that each solution corresponded to a prominent light source, such that the orientation of the light source was perpendicular to a plane tangent to the sphere at the corresponding solution coordinates. Since it was possible to establish the

³⁹ A maximum of four in this case.

⁴⁰ e.g. the occluding edges of the ping-pong ball.

surface orientation at this point using equations 85–86, it was also possible to compute the unit vector $\hat{n} = [-p, -q, 1]$, normal to this plane, which described the orientation of the light source relative to the image view-plane.

Note, in practical terms, this scheme did not take into account the contribution of multiple light sources to a single solution point. As well, care needed to be taken to ensure that no portion of the test image was saturated, as a search for an overall intensity maxima would not necessarily yield a unique solution under such circumstances. The dual solution intensity maximas shown in Figure 5.78 have been deliberately saturated to highlight the overall intensity maximas in the ping-pong ball image.

5.2.2 Reflectance Map Experimentation

B.K.P Horn describes in [12] the theoretical framework for the shape from shading paradigm. In the context of facial shape recovery, the inter-relationship between apparent image brightness and surface orientation is of particular interest. Horn demonstrated that to recover shape from shading, the following (minimal) information is required:

- I. The viewer direction or orientation in reference to the object being analyzed.
- II. The direction or orientation of light source(s) in reference to this object.
- III. A model or mathematical representation which can be used to describe the reflectance properties of the object's surface.

Given a situation where the facial pose was frontal, the viewer direction was known; the image view-plane was parallel to the subject's face. Nonetheless, implementation of a shape from shading technique in the context of facial recognition would, at minimum, require *a priori* knowledge of the light sources and a model which approximates the reflectance properties of various skin types. Clearly the most challenging problem is in modelling the reflectance properties of human skin. Such a model would have to account for attributes such as:

- I. Skin coloring (including seasonal tanning and effects of sunburn)
- II. Skin texture (eg. smooth, wrinkled)
- III. Natural-attributes (eg. sweat, oil, scars, facial hair, freckles, moles)
- IV. Non-natural attributes (eg. cosmetics, dirt).

Ideally, one would have advance knowledge the function $R(p, q)$, often referred to by Horn as a reflectance map, describing the relationship between apparent image brightness and surface orientation. To obtain a reflectance map for the ping pong ball image, the region of the test image containing the ball was analyzed to produce a quantized reflectance map in the form of a lookup table. For each point on the ball, the surface orientation was computed and the corresponding local brightness was recorded in the reflectance map, as shown in equation 87.

$$R(p, q) = [I_{x,y} \mid x^2 + y^2 \leq r^2], \text{ for } x, y = -r, \dots, r \quad (87)$$

A lookup table depicting a crude reflectance map is shown in Figure 5.80. Alternatively, when locating prominent light sources, it was found one could first construct the reflectance map, and then search it for intensity maximas instead. Regardless of the approach, it would appear shape information can be recovered from a facial image using experimentally obtained knowledge of the lighting sources in combination with one or more experimentally obtained skin-reflectance maps.

Although derivation of a general $R(p, q)$ for human skin seems highly improbable in the near future, it is expected a simple reflectance map for several generalized skin types could be constructed for a small population of subjects through the analysis of nasal image subregions; using Horn's techniques. In Chapter 3.8 it was shown the region of a facial image containing a human nose could be isolated⁴¹ reliably and in 3.9 techniques were described for predicting the location of the tip of the nose with a reasonable degree of accuracy. To engage the shape from shading techniques described by Horn, the surface orientation at an initial starting point⁴² for the solution must be known. Fortunately, the

⁴¹ eg. bounded by the coordinates $\{ \rho(w(lb_n), w(tb_n)), \rho(w(rb_n), w(bb_n)) \}$.

⁴² or the immediate surrounding area.

surface orientation at the nose-tip is known; one need only assume $(p, q) = (0, 0)$, so the plane tangent to the nose-tip is parallel to the image view-plane. The tip of the nose would represent an ideal initial solution point for the shape from shading algorithm. In fact, Horn illustrated⁴³ that a shape from shading model could be used to recover basic shape information from a nose.

It should be noted many of the algorithms and techniques presented in earlier discussion lend themselves to real-time implementation primarily because of their simplicity and minimal space requirements. Facial shape information would greatly enhance the recognition capabilities of a human identification system, although the time and space complexities for a shape from shading algorithm would undoubtedly be substantial. A practical and comprehensive solution with respect to facial shape recovery does not yet exist and even a partial implementation would be *non-trivial*. Nonetheless, this area has enormous intuitive appeal.

⁴³ Figure 11-7 on p 255 of [12] shows a contour map obtained using this technique superimposed on a human nose. It would appear as if a single light source perpendicular to the image view plane was used.

Figure 5.78 A three-dimensional intensity mesh of the ping-pong ball image used for reflectance map experimentation.

Figure 5.79 Intensity contours and light source points found on the ping-pong ball.

Figure 5.80 A reflectance map represented as a lookup table, showing a possible single point light source.

Chapter 6

CONCLUSIONS

This thesis has demonstrated the feasibility of implementing a computer vision system capable of identifying human facial images in a relatively unconstrained setting. It is still not precisely known how humans recognize or process facial images, however, this topic has been the focus of substantial amount of research in the area of perception and psychology. Facial identification by both man and machine is possible because it is assumed no two faces are identical. The task of facial identification can be carried out very rapidly and reliably by humans; who either do it subconsciously or find the conscious process of identification trivial. Machines, on the other hand, are not well suited to this type of problem. The perception and psychology literature suggests human facial identification is usually carried out in a single massively parallel operation instead of a serial operation based on individual features. In [7], Harmon points out "humans may name and use specific features on demand, but may not use them in the normal perceptual process." Conversely, the techniques best suited for machine implementation are those based on precise measurement and quantification of individual discrete fiducial features. Consequently a large section of this thesis dealt with the isolation, representation and classification of these features.

The research described here contributes to the area of human identification primarily by introducing an geometric facial model as well as feature extraction and classification algorithms suitable for both partially and fully automated identification of faces. Another

goal was to relax or eliminate some of the processing constraints which had appeared in the previous literature. For example, arbitrary backgrounds could be used, there were no special lighting requirements, no use of external clues or devices and no need for multiple training images. As well, extracted facial data was not dependent on the distance between the video camera and the subject. The main focus of this work was on algorithms which exploited the geometric structure and symmetrical aspects of a human face. These algorithms were particularly well suited towards real-time implementation and were capable of extracting geometric fiducial points and boundaries from an image scene containing a face. The information obtained was encoded in the form of a feature vector which could be used for classification or identification purposes. A method for computing the projection of certain facial features was also developed. As this thesis examined certain aspects of interactive or partially automated facial identification, prototypes for two scenarios were developed. The first was fully automated, while the others required a certain degree of human interaction. The automated facial extraction process was decomposed into several steps. If an arbitrary background scene was used, background calibration of the system was first required, using an image subtraction algorithm. Next, the subject's head was isolated from the background component of the image, and the occluding and outer boundaries of the subject's head were established. Various fiducial features were then isolated and quantified and distance ratios between the features were encoded in a multi-dimensional feature vector. Feature vectors were then added to a feature file for classification purposes, or compared with those in the feature file for identification purposes. The semi-automated implementations required that individual facial features be selected manually, but used exactly the same identification mechanism as in the fully automated case. Identification of individuals was based on a "nearest-neighbor" approach — accomplished by computing pairwise normalized Euclidian distances between an unknown feature vector and those in the classified set.

Due to the complex structure of facial images, identification techniques based on well established theoretical foundations have yet to be developed. However, many of the heuristically developed algorithms in this thesis demonstrated that highly specialized

techniques and procedures could be used to obtain encouraging results in a relatively unconstrained setting. Ultimately, techniques similar to the ones described here would lend themselves to implementation in a high-speed or real time environment. Although not explored here, it would appear that algorithms which exploit the facial symmetry about the vertical axis could be applied to each side of the face in a concurrent or parallel fashion.

Orientation proved to be the most challenging obstacle as well as the primary source of problems; all of the prototypes developed were sensitive to small deviations in orientation. Nonetheless, it is expected a facial shape recovery model could be developed to provide valuable orientation clues, as well as more detailed shape information, which would inevitably lead to improved consistency and greater identification accuracy.

References

- [1] L. Harmon, M. Kahn, R. Lasch, and P. Ramig, "Machine identification of human faces," *Pattern Recognition*, vol. 13, no. 2, pp. 97-110, 1981.
- [2] S. Cannon, G. Jones, and R. Campbell, "A computer vision system for the identification of individuals," in *IECON 1986 Conference Proceedings*, (Milwaukee, Wisconsin), pp. 347-351, September 1986.
- [3] L. Harmon and W. Hunt, "Automatic recognition of human face profiles," *Computer Graphics and Image Processing*, vol. 6, pp. 135-156, 1973.
- [4] J. Bradshaw and G. Wallace, "Models for the processing and identification of faces," *Perception and Psychophysics*, vol. 9, no. 5, pp. 443-448, 1971.
- [5] A. Goldstein and E. Mackenberg, "Recognition of human faces from isolated facial features: A developmental study," *Psychonomic Science*, vol. 6, no. 4, pp. 149-150, 1966.
- [6] A. Goldstein, L. Harmon, and A. Lesk, "Identification of human faces," *Proceedings of The IEEE*, vol. 59, pp. 748-760, May 1971.
- [7] A. Goldstein and L. Harmon, "Man-machine interaction in human face identification," *Bell Systems Technical Journal*, vol. 51, pp. 399-427, February 1972.
- [8] M. Rhodes, "Computer video image generation using identi-kit features," tech. rep., U.C.L.A., 1977.
- [9] R. Gonzales, "Image enhancement and restoration," in *Handbook of Pattern Recognition and Image Processing* (T. Young and K. Fu, eds.), pp. 191-213, Academic Press Inc., 1986.
- [10] W. Pratt, *Digital Image Processing*, ch. 17. NYC, New York: John Wiley and Sons, 1978.
- [11] L. Harmon, "The recognition of faces," *Scientific American*, vol. 229, pp. 71-82, November 1973.

- [12] B. Horn, *Robot Vision*, ch. 10-11. Cambridge, Massachusetts: The MIT Press, 1986.
- [13] L. Harmon, "Some aspects of recognition of human faces," in *Pattern Recognition in Biological and Technical Systems*, (New York), Springer-Verlag, 1971.
- [14] A. Jain and R. Dubes, *Algorithms For Clustering Data*, ch. 5. Prentice Hall Inc., 1988.
- [15] Y. Kaya and K. Kobayashi, "A basic study on human face recognition," in *Proceedings of the International Conference on Frontiers of Pattern Recognition*, (Hawaii), Academic Press Inc., 1971.
- [16] T. Sakai, M. Nagao, and M. Kidode, "Processing of multilevel pictures by computer: The case of photographs of human faces," in *Artificial Intelligence* (O. Firschein, ed.), pp. 219-226, AFIPS Press, Reston, Virginia, 1984.
- [17] J. Hochberg and R. Galper, "Recognition of faces: 1. an exploratory study," *Psychonomic Science*, vol. 9, no. 12, pp. 619-620, 1967.
- [18] MATROX Electronics Systems Limited, Dorval, Quebec, Canada, *PIP-512/1024A Hardware Manual; Video Digitiser Board for the IBM PC, XT and AT*, 238-a50-02/0 ed., March 1986.
- [19] MATROX Electronics Systems Limited, Dorval, Quebec, Canada, *MS DOS Software Library for the PIP-512/1024A Video Digitiser*, 238-a57-01/3 ed., February 1986.

Appendix A

Processing Environment

Software development and testing of the various facial identification programs was carried out on a Tandy 3000HD AT-compatible microcomputer, equipped with an Intel 80286 main processor and 80287 numeric co-processor, 640K of main memory and 2MB of extended memory, 60MB of fixed disk storage and a Matrox PIP-512 frame grabber board. The frame grabber was plugged into a bus expansion slot in the microcomputer and was capable of capturing frames in real time (eg. roughly 30 frames per second). Frame buffer dimensions were 512×512 pixels ×8 bits and 8 sets of input and RGB output lookup tables were available. Sixteen control and data registers were I/O mapped into the PC's address space, allowing for complete software control over the imaging board. Pixels could be accessed individually using the pixel address and data registers, or in a high-speed block-sequential manner using an on-board auto-increment mechanism. Frames were easily transferable to and from the board via memory or through some type of secondary storage device.

A Sony AVC-3260 video camera was used for image acquisition. The camera output an RS-170 video signal which also provided synchronization for the board. The signal was digitized by the on-board A/D converter to produce 8 bits of data which were passed to a 256-byte input lookup table,⁴⁴ which in turn mapped incoming pixel values into the frame buffer. Output from the frame buffer was first processed by a set of RGB output

⁴⁴ All lookup tables were represented as 256 byte vectors.

lookup tables and then by three D/A converters to produce the output video signal in real-time. An overview of the system is shown in Figure A.81. In addition, the imaging board had a Synertek Inc. SY6845E CRT controller which is not shown⁴⁵.

All system software was designed and implemented by the author using version 1.0 of Borland's Turbo-C compiler and version 4.0 of Microsoft's Macro Assembler, under version 3.10 of Microsoft's DOS operating system. The "large" C memory model, based on 32-bit pointers, was used. The machine was also equipped with a Phoenix Inc. BIOS. The system's imaging control software contained a specialized set of functions which were specifically designed for this type of application. Figure A.82⁴⁶ shows expected average pixel transfer times for the PIP-512. Device drivers written by the author made use of I/O block transfers in conjunction with the board's auto-increment mechanism to achieve near optimal performance as far as data transfers were concerned. Disk images occupied one quarter of a megabyte of memory and though the 640K of primary memory was certainly required, the available extended memory was not used directly by any of the system software.

	AT	PC or XT
DMA Block Transfers	5.3us	2.6us
I/O Block Transfers	1.8us	10.8us

Figure A.82 Matrox PIP-512 pixel transfer rates.

⁴⁵ A detailed technical summary for the PIP-512 can be found in "MATROX Electronics Systems Limited, Dorval, Quebec, Canada, *PIP-512/1024A Hardware Manual, Video Digitiser Board for the IBM PC, XT and AT*, 238-a50-02/0 ed., March 1986."

⁴⁶ From p.20 of "MATROX Electronics Systems Limited, Dorval, Quebec, Canada, *MS DOS Software Library for the PIP 512/1024A Video Digitiser* 238-a57-01/3 ed., February 1986."

Figure A.81 Overview of the system hardware configuration.

Appendix B

Source FACEMAP Point Data

The following tables contain FaceMAP data used in the prediction of features by the automated classification and recognition algorithm. This data was also used by the manual training and classification programs to produce known feature vector sets for 16 faces, with each set being composed of 4 or 9 features.

Normalized vertical distance from midpoint between the eyes Normalized horizontal distance from midpoint between the eyes Absolute vertical position (y) Absolute horizontal position (x)						
I.D.	Point#	Point Description				
Face#1	1	pupil of left eye	270.0	239.0	-0.500	0.013
	2	pupil of right eye	383.0	236.0	0.500	-0.013
	3	left corner (lefteye)	245.0	242.0	-0.721	0.040
	4	right corner (lefteye)	296.0	241.0	-0.270	0.031
	5	left corner (right eye)	353.0	240.0	0.235	0.022
	6	right corner (right eye)	405.0	238.0	0.695	0.004
	7	ip of the nose	334.0	290.0	0.066	0.465
	8	left side of the nose (@tip)	293.0	290.0	-0.296	0.465
	9	right side of the nose (@tip)	366.0	290.0	0.350	0.465
	10	left corner of the mouth	281.0	358.0	-0.403	1.066
	11	right corner of the mouth	374.0	359.0	0.420	1.075
Face#2	1	pupil of left eye	215.0	225.0	-0.500	-0.010
	2	pupil of right eye	317.0	227.0	0.500	0.010
	3	left corner (lefteye)	193.0	235.0	-0.716	0.088
	4	right corner (lefteye)	239.0	227.0	-0.265	0.010
	5	left corner (right eye)	294.0	230.0	0.275	0.039
	6	right corner (right eye)	342.0	235.0	0.745	0.088
	7	tip of the nose	267.0	287.0	0.010	0.598
	8	left side of the nose (@tip)	239.0	287.0	-0.265	0.598
	9	right side of the nose (@tip)	297.0	287.0	0.304	0.598
	10	left corner of the mouth	233.0	351.0	-0.324	1.225
	11	right corner of the mouth	303.0	351.0	0.363	1.225

```

/****
  calculate the normalization factor (eg. standard deviation)
  for each active feature
****/

for(i=0;i<FACTIVE;++i) {
    fsum = 0.0;
    for(x=0;x<MAXPERSONS;++x) {
        if(person[x].active) {
            diff = person[x].fvect[i] - mean[i];
            fsum = fsum + (diff*diff);
        } /* if */
    } /* for */
    diff = wp.fvect[i] - mean[i];
    fsum = fsum + (diff*diff);
    fsum = fsum / (double)(total+1);
    fvnormal[i] = sqrt(fsum);
} /* for */

/****
  compute the pairwise euclidian distances between the unknown
  feature vector and all those in the set of known feature
  vectors
****/

total=0;
for(x=0;x<MAXPERSONS;++x) {
    if(person[x].active) {
        fsum = 0.0;
        for(i=0;i<FACTIVE;++i) {
            diff = wp.fvect[i]-person[x].fvect[i];
            if(fvnormal[i]!=0.0) diff = diff / fvnormal[i];
            fsum = fsum + (diff*diff);
        } /* for */
        euclid = sqrt(fsum);
        fvdist[total]=euclid;
        fvindx[total]=x;
        ++total;
    } /* if */
} /* for */

```

Normalized vertical distance from midpoint between the eyes Normalized horizontal distance from midpoint between the eyes Absolute vertical position (y) Absolute horizontal position (x)						
I.D.	Point#	Point Description				
Face#6	1	pupil of left eye	182.0	234.0	-0.500	0.011
	2	pupil of right eye	276.0	232.0	0.500	-0.011
	3	left corner (lefteye)	158.0	243.0	-0.755	0.106
	4	right corner (lefteye)	208.0	236.0	-0.223	0.032
	5	left corner (right eye)	254.0	237.0	0.266	0.043
	6	right corner (right eye)	302.0	239.0	0.777	0.064
	7	tip of the nose	224.0	296.0	-0.053	0.670
	8	left side of the nose (@tip)	195.0	296.0	-0.362	0.670
	9	right side of the nose (@tip)	260.0	296.0	0.330	0.670
	10	left corner of the mouth	191.0	368.0	-0.404	1.436
	11	right corner of the mouth	268.0	367.0	0.415	1.426
Face#7	1	pupil of left eye	256.0	226.0	-0.500	0.018
	2	pupil of right eye	339.0	223.0	0.500	-0.018
	3	left corner (lefteye)	237.0	230.0	-0.729	0.066
	4	right corner (lefteye)	275.0	230.0	-0.271	0.066
	5	left corner (right eye)	319.0	229.0	0.259	0.054
	6	right corner (right eye)	359.0	227.0	0.741	0.030
	7	tip of the nose	299.0	270.0	0.018	0.548
	8	left side of the nose (@tip)	273.0	270.0	-0.295	0.548
	9	right side of the nose (@tip)	326.0	270.0	0.343	0.548
	10	left corner of the mouth	268.0	341.0	-0.355	1.404
	11	right corner of the mouth	327.0	340.0	0.355	1.392
Face#8	1	pupil of left eye	215.0	205.0	-0.500	0.023
	2	pupil of right eye	303.0	201.0	0.500	-0.023
	3	left corner (lefteye)	194.0	209.0	-0.739	0.068
	4	right corner (lefteye)	235.0	208.0	-0.273	0.057
	5	left corner (right eye)	283.0	205.0	0.273	0.023
	6	right corner (right eye)	322.0	203.0	0.716	0.000
	7	tip of the nose	263.0	257.0	0.045	0.614
	8	left side of the nose (@tip)	232.0	257.0	-0.307	0.614
	9	right side of the nose (@tip)	290.0	257.0	0.352	0.614
	10	left corner of the mouth	228.0	314.0	-0.352	1.261
	11	right corner of the mouth	296.0	314.0	0.420	1.261

Normalized vertical distance from midpoint between the eyes
 Normalized horizontal distance from midpoint between the eyes
 Absolute vertical position (y)
 Absolute horizontal position (x)

I.D.	Point#	Point Description				
Face#9	1	pupil of left eye	260.0	252.0	-0.500	0.027
	2	pupil of right eye	334.0	248.0	0.500	-0.027
	3	left corner (lefteye)	242.0	257.0	-0.743	0.095
	4	right corner (lefteye)	276.0	253.0	-0.284	0.041
	5	left corner (right eye)	317.0	254.0	0.270	0.054
	6	right corner (right eye)	353.0	251.0	0.757	0.014
	7	tip of the nose	306.0	294.0	0.122	0.595
	8	left side of the nose (@tip)	279.0	294.0	-0.243	0.595
	9	right side of the nose (@tip)	332.0	294.0	0.473	0.595
	10	left corner of the mouth	273.0	353.0	-0.324	1.392
	11	right corner of the mouth	336.0	351.0	0.527	1.365
Face#10	1	pupil of left eye	165.0	261.0	-0.500	-0.005
	2	pupil of right eye	257.0	262.0	0.500	0.005
	3	left corner (lefteye)	144.0	261.0	-0.728	-0.005
	4	right corner (lefteye)	185.0	262.0	-0.283	0.005
	5	left corner (right eye)	236.0	264.0	0.272	0.027
	6	right corner (right eye)	277.0	268.0	0.717	0.071
	7	tip of the nose	209.0	319.0	-0.022	0.625
	8	left side of the nose (@tip)	183.0	319.0	-0.304	0.625
	9	right side of the nose (@tip)	235.0	319.0	0.261	0.625
	10	left corner of the mouth	174.0	392.0	-0.402	1.418
	11	right corner of the mouth	241.0	389.0	0.326	1.386
Face#11	1	pupil of left eye	230.0	253.0	-0.500	-0.007
	2	pupil of right eye	305.0	254.0	0.500	0.007
	3	left corner (lefteye)	215.0	254.0	-0.700	0.007
	4	right corner (lefteye)	244.0	259.0	-0.313	0.073
	5	left corner (right eye)	287.0	261.0	0.260	0.100
	6	right corner (right eye)	319.0	256.0	0.687	0.033
	7	tip of the nose	265.0	305.0	-0.033	0.687
	8	left side of the nose (@tip)	247.0	305.0	-0.273	0.687
	9	right side of the nose (@tip)	286.0	305.0	0.247	0.687
	10	left corner of the mouth	239.0	361.0	-0.380	1.433
	11	right corner of the mouth	296.0	361.0	0.380	1.433

Normalized vertical distance from midpoint between the eyes Normalized horizontal distance from midpoint between the eyes Absolute vertical position (y) Absolute horizontal position (x)						
I.D.	Point#	Point Description				
Face#12	1	pupil of left eye	195.0	250.0	-0.500	-0.005
	2	pupil of right eye	293.0	251.0	0.500	0.005
	3	left corner (lefteye)	176.0	255.0	-0.694	0.046
	4	right corner (lefteye)	219.0	255.0	-0.255	0.046
	5	left corner (right eye)	270.0	257.0	0.265	0.066
	6	right corner (right eye)	311.0	250.0	0.684	-0.005
	7	tip of the nose	243.0	312.0	-0.010	0.628
	8	left side of the nose (@tip)	216.0	312.0	-0.286	0.628
	9	right side of the nose (@tip)	272.0	312.0	0.286	0.628
	10	left corner of the mouth	209.0	382.0	-0.357	1.342
	11	right corner of the mouth	281.0	385.0	0.378	1.372
Face#13	1	pupil of left eye	228.0	225.0	-0.500	-0.006
	2	pupil of right eye	308.0	226.0	0.500	0.006
	3	left corner (lefteye)	212.0	231.0	-0.700	0.069
	4	right corner (lefteye)	246.0	226.0	-0.275	0.006
	5	left corner (right eye)	291.0	226.0	0.287	0.006
	6	right corner (right eye)	322.0	231.0	0.675	0.069
	7	tip of the nose	271.0	265.0	0.037	0.494
	8	left side of the nose (@tip)	243.0	265.0	-0.312	0.494
	9	right side of the nose (@tip)	296.0	265.0	0.350	0.494
	10	left corner of the mouth	237.0	330.0	-0.388	1.306
	11	right corner of the mouth	301.0	325.0	0.412	1.244
Face#14	1	pupil of left eye	180.0	273.0	-0.500	0.035
	2	pupil of right eye	265.0	267.0	0.500	-0.035
	3	left corner (lefteye)	160.0	280.0	-0.735	0.118
	4	right corner (lefteye)	197.0	275.0	-0.300	0.059
	5	left corner (right eye)	246.0	277.0	0.276	0.082
	6	right corner (right eye)	283.0	273.0	0.712	0.035
	7	tip of the nose	219.0	312.0	-0.041	0.494
	8	left side of the nose (@tip)	193.0	312.0	-0.347	0.494
	9	right side of the nose (@tip)	252.0	312.0	0.347	0.494
	10	left corner of the mouth	197.0	381.0	-0.300	1.306
	11	right corner of the mouth	254.0	381.0	0.371	1.306

Normalized vertical distance from midpoint between the eyes Normalized horizontal distance from midpoint between the eyes Absolute vertical position (y) Absolute horizontal position (x)						
I.D.	Point#	Point Description				
Face#15	1	pupil of left eye	205.0	296.0	-0.500	0.017
	2	pupil of right eye	293.0	293.0	0.500	-0.017
	3	left corner (lefteye)	185.0	300.0	-0.727	0.062
	4	right corner (lefteye)	224.0	299.0	-0.284	0.051
	5	left corner (right eye)	273.0	298.0	0.273	0.040
	6	right corner (right eye)	309.0	301.0	0.682	0.074
	7	tip of the nose	252.0	338.0	0.034	0.494
	8	left side of the nose (@tip)	221.0	338.0	-0.318	0.494
	9	right side of the nose (@tip)	282.0	338.0	0.375	0.494
	10	left corner of the mouth	220.0	406.0	-0.330	1.267
	11	right corner of the mouth	284.0	404.0	0.398	1.244
Face#16 (Subject#1)	1	pupil of left eye	193.0	184.0	-0.500	0.008
	2	pupil of right eye	319.0	182.0	0.500	-0.008
	3	left corner (lefteye)	168.0	193.0	-0.698	0.079
	4	right corner (lefteye)	224.0	189.0	-0.254	0.048
	5	left corner (right eye)	284.0	190.0	0.222	0.056
	6	right corner (right eye)	344.0	190.0	0.698	0.056
	7	tip of the nose	258.0	256.0	0.016	0.579
	8	left side of the nose (@tip)	213.0	256.0	-0.341	0.579
	9	right side of the nose (@tip)	298.0	256.0	0.333	0.579
	10	left corner of the mouth	204.0	356.0	-0.413	1.373
	11	right corner of the mouth	303.0	358.0	0.373	1.389

Appendix C

Source Listing of REC.C

The following C source code listing is of the primary recognition function (FV_euclid) used by all of the facial identification programs. The data structures used for storing records in the feature file are shown here as well (all though none of the feature file I/O functions have been included.) This code fragment also serves as an example of the implementation form taken by virtually all of the facial identification functions and procedures.

```

/*****
Module: REC.C - recognition module for facial identification procedures.

Contents: Global data structures used by all facial identification
procedures and the function FV_euclid() which is used to carry
out the pairwise euclidian distance calculations between the set
of known feature vectors and one that is unknown. File i/o and
general feature file maintenance functions are not included here.

Date Written: May 1988
Last Revised: June 1989

                                                                 Ian MacLaren
*****/

/* header/include files */

#include <stdio.h>          /* standard i/o */
#include <fcntl.h>          /* file control */
#include <math.h>           /* math definitions */
#include <stdlib.h>         /* standard library */

#define MAXFEATURES 15     /* maximum number of features per record */
#define MAXPERSONS 100    /* maximum number of subjects in feature file */

```

```

#define MAXIPDAT      06      /* maximum image proc data elements/rec */
#define MAXNAME       40      /* maximum length of name/subject i.d. string */
#define MAXCOMMENT    80      /* maximum length of comment string */
#define FACTIVE       9       /* current number of active features */

#define TRUE          01      /* boolean true */
#define FALSE         00      /* boolean false */

/* data structure for a single record in the feature file */

struct face {
    short int id;             /* record identification # */
    short int active;         /* record activation flag */
    short int delflag;        /* record deletion flag */
    char name[MAXNAME];       /* subject's name or identifying string */
    short int day;            /* day of classification */
    short int month;          /* month of classification */
    short int year;           /* year of classification */
    char comment[MAXCOMMENT]; /* general comments or items of interest */
    double ipdat[MAXIPDAT];   /* image processing data storage */
    double fvect[MAXFEATURES]; /* feature vector data */
}; /* face */

/* globally visible data structures */

struct face
    wp,                       /* storage for unknown individual */
    person[MAXPERSONS];       /* storage for all known subjects */

double fvnormal[MAXFEATURES]; /* vector of feature normals */
double fvdist[MAXPERSONS];    /* vector of euclidian measurements */
short int fvindx[MAXPERSONS]; /* vector of feature indices */

short int
    numactive,                /* number of active records in feature file */
    numtotal,                 /* total number of records in feature file */
    fvtotal;

FILE *facefile;               /* file handle for the feature file */
short int faces;              /* file descriptor for the feature file */
short int lostat;             /* global file i/o status */

```

```

/*****
FV_EUCLID - recognition function for facial identification procedures
*****/

FV_euclid()
{
    short int
        total,          /* total number of individuals */
        x,y,i,          /* index variables */
        done,           /* sorting control variable */
        sort,           /* sort index */
        tempndx;         /* temporary index */
    double
        fsum,           /* sum storage */
        diff,           /* difference storage */
        euclid,         /* euclidian distance */
        tempfv,         /* temporary variable */
        mean[MAXFEATURES]; /* mean vector for all features */

    /****
    upon entry, the global data structure "wp" must contain the raw
    feature data for the unknown individual. As well, if the set of
    feature vectors for individuals known to the system is not present
    in global memory, then retrieve them from the feature file.
    *****/

    if(!readflag) read_features();

    /****
    calculate mean for each active feature
    *****/

    for(i=0;i<FACTIVE;++i) {
        total=0;
        fsum = 0.0;
        for(x=0;x<MAXPERSONS;++x) {
            if(person[x].active) {
                fsum = fsum + person[x].fvect[i];
                ++ total;
            } /* if */
        } /* for */
        fsum = fsum + wp.fvect[i];
        mean[i]=fsum/(double) (total+1);
    } /* for */

```

```

/****
  calculate the normalization factor (eg. standard deviation)
  for each active feature
****/

for(i=0;i<FACTIVE;++i) {
  fsum = 0.0;
  for(x=0;x<MAXPERSONS;++x) {
    if(person[x].active) {
      diff = person[x].fvect[i] - mean[i];
      fsum = fsum + (diff*diff);
    } /* if */
  } /* for */
  diff = wp.fvect[i] - mean[i];
  fsum = fsum + (diff*diff);
  fsum = fsum / (double)(total+1);
  fvnormal[i] = sqrt(fsum);
} /* for */

/****
  compute the pairwise euclidian distances between the unknown
  feature vector and all those in the set of known feature
  vectors
****/

total=0;
for(x=0;x<MAXPERSONS;++x) {
  if(person[x].active) {
    fsum = 0.0;
    for(i=0;i<FACTIVE;++i) {
      diff = wp.fvect[i]-person[x].fvect[i];
      if(fvnormal[i]!=0.0) diff = diff / fvnormal[i];
      fsum = fsum + (diff*diff);
    } /* for */
    euclid=sqrt(fsum);
    fvdist[total]=euclid;
    fvindx[total]=x;
    ++total;
  } /* if */
} /* for */

```

```

/****
Sort pairwise difference and index vectors in ascending order.
Afterwards, the individual @ person[fvindx[0]] is the system's
primary choice.
****/

done=FALSE;
for(sort=0; (sort<total)&&(done==FALSE); ++sort) {
    done=TRUE;
    for(x=0; x<total-1; ++x) {
        if(fvdist[x]>fvdist[x+1]) {
            /* switch elements */
            tempfv = fvdist[x];
            fvdist[x]=fvdist[x+1];
            fvdist[x+1] = tempfv;
            tempndx = fvindx[x];
            fvindx[x] = fvindx[x+1];
            fvindx[x+1] = tempndx;
            done=FALSE;
        } /* if */
    } /* for */
} /* for */
fvtotal=total;

} /* fv euclid */

```

END

2010 2/9 1

FIN