

THE FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

AUTOMATED FACE TRACKING AND RECOGNITION

By

MATTHEW CURTIS HESHER

A Thesis submitted to the
Department of Computer Science
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Summer Semester, 2003

The members of the Committee approve the dissertation of Matthew Curtis Heshel defended on April 10 2003.

Gordon Erlebacher
Professor Directing Thesis

Anuj Srivastava
Professor Co-Directing Thesis

Kyle Gallivan
Committee Member

Approved:

Sudhir Aggarwal, Chair
Department of Computer Science

The Office of Graduate Studies has verified and approved the above named committee members.

ACKNOWLEDGEMENTS

The algorithms proposed in chapters 3 and 4 of this thesis are part of a collaborative effort between the Department of Statistics and Computational Science and Information Technology (CSIT) to develop efficient algorithms for automated target tracking and recognition. This work could not have been accomplished without the assistance and guidance of many people, including, but not limited to, Dr. Anuj Srivastava (Dept. of Statistics, Florida State University) and Dr. Gordon Erlebacher (CSIT, Florida State University).

I would also like to acknowledge everyone who has helped me in my educational journey including my grade school science and reading teachers, my undergraduate mathematics professors, and my advisor who has provided many opportunities for learning and growth. I would especially like to thank my parents for their continued support throughout my educational experience.

The following generous grants partially supported the research presented in this thesis: ARO Grant DAAD19-99-1-0267 and NSF Grant DMS-0101429.

TABLE OF CONTENTS

List of Tables	vi
List of Figures	vii
Abstract	ix
1. A REVIEW OF FACE TRACKING AND RECOGNITION	1
1.1 Introduction	1
1.2 Difficulties in Face Tracking and Recognition	2
1.2.1 Tracking	2
1.2.2 Recognition	3
1.3 Current Methods for Face Recognition	4
1.3.1 Past Research	4
1.3.2 A Summary of Survey Papers	8
1.4 Commercial Implementations in Automated Face Tracking and Recognition ..	9
1.5 Conclusions	11
2. DATA ACQUISITION AND REPRESENTATION	12
3. FACE TRACKING USING IMAGES GENERATED FROM GEOMETRY	15
3.1 Algorithm	15
3.2 Experiment and Results	19
3.3 Conclusions and Future Work	23
4. FACE RECOGNITION USING RANGE IMAGES	24
4.1 Representation of Facial Surfaces	25
4.2 Data Acquisition	26
4.3 Generation of Range Images	27
4.4 Planar Registration of Range Images	30
4.5 Preprocessing of Range Images	32
4.6 Dimension Reduction	34
4.7 Identification	37
4.8 Experiment and Results	37
4.9 Conclusions and Future Work	41

5. TOOLS FOR DERIVING A STATISTICAL MODEL OF RANGE IMAGES	43
APPENDIX A: Derivation of $\frac{\partial}{\partial q_i} C(\mathbf{Q})$	46
REFERENCES	47
BIOGRAPHICAL SKETCH	55

LIST OF TABLES

4.1 Each row in this table indicates the results of an experiment in identification using range images and PCA. The first column shows the number of training images used; the second column shows how many images were used in the test data set; the third column lists the percentage of correctly identified faces. In all experiments there is no intersection between the training and test data sets.	38
4.2 In these results 30 eigenvectors from PCA were used for projection with range images of size 242×347 (i.e., where 10 eigenvectors have previously been used for projection, we now use 30, and where range images were previously 41×57 pixels in size, they are now 242×347 .)	38
4.3 In these results 60 eigenvectors from PCA were used for projection with range images of size 242×347 .	39
4.4 For this experiment, 82 subjects with a total of 492 faces are used. Images are 41×61 pixels. All results are percentages. PCA (Principal Component Analysis), ICA (Independent Component Analysis) and FDA (Fisher's Discriminate Analysis) are compared. Projections are 10, 30, or 60 dimensional as indicated. Algorithmic limitations lead to the missing results.	39
4.5 For this experiment, 82 subjects with a total of 492 faces are used. Images are 202×306 pixels. All results are percentages. PCA (Principal Component Analysis), ICA (Independent Component Analysis) and FDA (Fisher's Discriminate Analysis) are compared. Projections are 10, 30, or 60 dimensional as indicated. Computational and algorithmic limitations lead to missing results.	39

LIST OF FIGURES

2.1	An illustration [52] of the measuring principle used by the Minolta Vivid 700 camera. By passing a laser light through a cylindrical lens, a horizontal light-stripe is created. The light-stripe is then reflected by a galvano mirror onto objects in the imaged scene. A CCD then receives the reflected light-stripe and generates distance information by triangulation. The galvano mirror is then rotated resulting in a projection of the light-stripe onto a different part of the scene and the measuring begins again. This is done 200 times for one range scan.	13
2.2	Facial meshes captured using the Minolta Vivid 700 3D camera. These meshes have been decimated for illustration purposes to 1,000 triangles each down from a typical 15,000 triangles.	14
3.1	An illustration of the basic flow of the tracking algorithm. The mesh corresponding to the object to track is manually chosen.	16
3.2	(a) A rendered image of a face displayed over the rendering axis. (b) A rendering of the assumed position, orientation, and scale of a face in the first frame of the video.	17
3.3	Pictures of a synthetically generated box and a scanned face. Textures for the box were acquired using a digital camera and mapped to a hand generated box manually. The mesh and texture of the face was acquired automatically using a Minolta Vivid 700 range camera.	19
3.4	(a) Two synthetic target images with Gaussian noise. Directly below each target image, part (b) shows the corresponding best estimates to the target’s position and orientation.	20
3.5	An illustration of the perspective projection concept [13]. This projection was used in the generation of the synthetic target tracking video for the box.	21
3.6	Two target frames from a video captured using a Sony XC-003 digital frame grabber are shown (labeled “Target”). The apparent shroud is a black shirt used to lessen background clutter. Gaussian noise ($N(0, 0.5)$) was added to the targets to simulate poor video clarity. The image labeled as “Estimate” shows the corresponding face mesh rendered at a position and orientation where $C(\mathbf{x}, \boldsymbol{\theta}) < \beta$ (see section 3.1). The image labeled “Mesh” shows the textureless mesh (at a reduced resolution) used for the conditional estimates shown in the image labeled “Estimate”. The image labeled “Difference” is a visualization of $\mathbf{T}_k - \mathbf{E}(\mathbf{x}, \boldsymbol{\theta})$ for the estimate shown above it.	22
4.1	An example data capture session. Subjects stay in a predetermined position and orientation.	26

4.2	An illustration of the Line Crossing Algorithm. The point in question, P , is outside the triangle (exterior). This graphic is borrowed from [1].	28
4.3	Facial range images. The top row of images displays three, unique persons with the same expression. The bottom row of images displays the same person under three different facial expressions. Aside from the background, light areas of range images indicate portions of the face further from the range camera while darker areas indicate portions closer to the range camera.	29
4.4	In these range images, the asterisk indicates the pixel of <i>strongest</i> intensity. The line extending from the asterisk toward the forehead indicates the pixel of strongest intensity for a number of rows. The image pairs illustrate range images before rotational correction (left) and after (right).	31
4.5	This mask is used to crop range images so that every range image has data in corresponding pixels. This is the second of three preprocessing steps before the use of dimension reduction techniques.	33
4.6	Range images generated from a facial mesh. The black dot in the left image represents a hole in the range image. The same range image is displayed on the right after hole patching. Lighter pixel values indicate parts of the face further from the camera while darker pixel values indicate parts of the face closer to the camera. Completely black areas indicate no data. Note that the nose is dark gray, but not black. This is the final preprocessing step before the use of dimension reduction techniques.	33
4.7	The first 20 eigenvalues plotted along the horizontal axis in order of decreasing value.	35
4.8	Three eigenvectors associated with the three largest eigenvalues in descending order from left to right. Lighter pixel values indicate regions of greater variation and darker pixel values indicate regions of less variation.	36
4.9	This dendrogram illustrates the Euclidean distance between the projections of twelve faces in the data set. Faces one through six belong to one person; faces seven through twelve belong to a different person. Using this method, we see that face eight is relatively far from both groups.	40
4.10	On the left is the range image of the person we wish to identify (person 5, expression 5). On the right is the incorrect result of the nearest-neighbor matching algorithm (person 12, face 2). This mismatch occurred when using three training faces per person.	41
5.1	(a) An illustration of the vector from a view point to a location in the scene. Intersections of the tree object in the scene with the vector are of interest. (b) The tree mesh used in the generation of the intersection maps shown in figure 5.2.	44
5.2	Two intersection maps.	45

ABSTRACT

We have considered the problem of tracking and recognition using a three dimensional representation of human faces. First we present a review of the research in the tracking and recognition fields including a list of several commercially available face tracking and recognition systems. Next, two algorithms are described: one for tracking faces from observed images and one for recognition of faces from observed geometries. The tracking algorithm uses 3D shape and texture of a human face to estimate the changing position and orientation of a real face in a video image sequence. The recognition algorithm uses principal component analysis (PCA) of range images generated from the 3D shape of a human face to create a database of low-dimensional face representations for efficient recognition. Range images are robust to illumination and texture variations and thus avoid some of the current limitations in face recognition.

CHAPTER 1

A REVIEW OF FACE TRACKING AND RECOGNITION

1.1 Introduction

Face tracking and recognition are important parts of our daily lives. Tracking an object under time varying position and orientation is a basic ability of the human visual system ([18], pp. 53). Studies ([18], pp. 53-54) show that infants are, in some way, preprogrammed to recognize and pay attention to faces more than other objects. Throughout our lives people present their face as a form of identification in person and through the use of photo identification cards such as a driver's license or passport. Since face identification is so pervasive in the natural world, it is reasonable to consider faces as a means for recognition using machines. So far, robust algorithms to perform automated face tracking and recognition in unconstrained environments have not been achieved. To further complicate matters, psychology suggests that the principal means of identification used by humans changes from a primarily feature-based method in childhood to a primarily holistic-based method in adulthood ([18], pp. 28-29). Which method, if either, will work best in an automated recognition system?

A great wealth of research has been done in many fields to determine how to best track and recognize faces, how to simulate (or surpass) human face tracking and recognition performance [30], and how to overcome difficulties that hinder the development of automated face tracking and recognition. There are several imaging modalities: video, infrared, and

three dimensional (3D) scanning. Although there has been tremendous research in video, other modalities deserve attention.

Our goal is to use the information available in 3D representations of the face in the development of robust face tracking and recognition algorithms. In the remaining sections of chapter 1 we discuss some of the difficulties, research, and commercial applications in automated tracking and recognition. In chapter 2, a data acquisition method and data representation is explained. The techniques explained in chapter 2 are then used for the tracking and recognition algorithms presented in chapters 3 and 4 respectively. A brief overview of a statistical model for range images is then given in chapter 5.

1.2 Difficulties in Face Tracking and Recognition

There are many challenges that impede the development of effective solutions to the face tracking and recognition problem. They include poor image quality, obscuration, deformations, clutter, texture variation, and many more. We describe some of them in this section. We adopt the classification system presented in [18], based on extrinsic and intrinsic variations. Variation in images of faces due to external factors, such as lighting and occlusions, is called extrinsic while variation in these same images due to internal factors, such as deformations of the facial surface from expression, is called intrinsic.

1.2.1 Tracking

There are three fundamental extrinsic difficulties with target tracking [64]:

1. Many parameters including projection, occlusion, and reflection affect the observed mesh in an image. The relationship between these image parameters and images of a mesh (see chapter 2) rendered in a scene is nonlinear.
2. The matrix that gives the orientation of a mesh in a scene is not unique (i.e., rotations of 90 and -270 degrees are identical).

3. The dimensionality of the observation space (the number of pixels in an image) is usually much greater than the number of variables computed from that space. Therefore the estimated parameters frequently rely on a subset of the pixels in the image. When the target is in motion this subset depends on time.

Other obstacles to reliable tracking include image segmentation ([56], sec. 3 & 7) and, in some instances, target identification [64], which are themselves active fields of research.

1.2.2 Recognition

Many issues hinder research efforts in the field of face recognition. Variation exists in every imaging modality used, and finding fast, simple algorithms that are robust to variation is difficult (as evidenced by years of research). Categorizing the variation may be helpful in the development of effective face recognition algorithms. Intrinsic sources of variation include identity, facial expression, speech, gender, and age [18]. Extrinsic sources of variation include viewing geometry, illumination, imaging processes, and other objects. Viewing geometry includes pose changes, either by the observer or the object to be recognized; illumination changes include shading, color, self-shadowing, and specular highlights; imaging process variations include resolution, focus, imaging noise, sampling technique, and perspective distortion effects; variation from other objects include occlusions, shadowing, and indirect illumination. These sources of variation may or may not hinder the recognition process depending on which algorithm is used. It is possible that the variation due to factors such as facial expression, lighting, occlusions, and pose is larger than the variation due to identity [12, 18]. That makes identification under such varying environments a difficult task. However, human proficiency at face recognition [30] has motivated enormous research in this area despite these challenges. (The ability of humans to recognize faces is also an actively researched field with widely varying results depending on numerous factors. Additional information on this topic can be found predominantly in the psychology literature [6, 7].)

1.3 Current Methods for Face Recognition

Face recognition is an interdisciplinary area of research. Researchers contributing to this area come from diverse backgrounds including, but not limited to, electrical engineering, mathematics, statistics, physics, and computer science. This section includes a classification scheme along with some significant contributions in the field of face recognition.

There are currently many different approaches to face recognition. We classify them as either geometrical or nongeometrical [Dr. Anuj Srivastava, personal communication]. Geometrical approaches, such as the algorithm for face recognition described in chapter 4, use the physical geometry of the face coded as a mesh (see chapter 2 for an explanation of meshes). Recognition is then performed by deriving geometric characteristics from the face (such as curvature data). This is fundamentally different from nongeometrical approaches. These begin with two dimensional (2D) face images. Features are then extracted from these images for recognition.

1.3.1 Past Research

Most modalities are image-based. They may be classified into three categories: statistical, neural, or feature-based [11]. Statistical approaches to face recognition typically use linear and nonlinear discrimination techniques to reduce the dimensionality of the image space to some lower (and therefore more computationally tractable) subspace. Examples of statistical approaches include eigenfaces [53], Kernel Discriminant Analysis (KDA) [45], and the Support Vector Machine (SVM) [37].

Principal Component Analysis (PCA), a statistical approach based on eigendecomposition, was developed independently by Pearson (1901) [36] and Hotelling (1933) [31]. It is a linear transformation that removes the correlation among elements of a random vector from a discrete sample. Karhunen and Loève (1947) [39, 48] developed a similar transformation for continuous signals (known as the Karhunen-Loève or K-L transform). In the literature, PCA and the (discrete) K-L transform (as well as the Hotelling transform) are regarded as the same technique ([10], pp. 297, [19], pp. 148) and the names are used interchangeably. PCA

extracts the most prominent variation between 2D images (typically gray scale) by redefining the basis of the image space so that the first axis (principal component) corresponds to the dimension of maximal variation in the data, the second axis corresponds to the maximal remaining variation in a dimension orthogonal to the first axis, etc. Sirovich and Kirby (1987) [62] created an application based on PCA to reduce the dimensionality of face images. This is possible because most of the variation in a set of face images can be captured by a relatively small number of principal components, e.g., the ten dominant principal components associated with the ten largest eigenvalues. By projecting a face image into these ten principal components a set of coefficients are generated that uniquely describe the image as a linear combination of the computed eigenfaces. In other words, the image is reduced from a very large space ($\mathfrak{R}^{128 \times 128}$ for a 128×128 image) to a much smaller space (\mathfrak{R}^{10}) with minimal loss of information. Using the concepts presented by Sirovich and Kirby [62], Pentland (1991) [72] performed an experiment in face recognition. They coined the term *eigenface* due to the “ghostly” face-like appearance of the principal components computed from the training data set of facial images.

O’Toole *et al.* [53] evaluate the usefulness of higher order eigenvectors generated from PCA for recognition. Higher order eigenvectors are associated with eigenvalues of lower magnitude. Let \vec{v}_k be the eigenvector corresponding to the eigenvalue λ_k where $\lambda_k \geq \lambda_{k+1}$. They conclude that eigenvectors \vec{v}_{50} to \vec{v}_{70} are the best for recognition while the lower order eigenvectors ($< \vec{v}_{50}$) are optimal in the least squares sense for appearance reconstruction.

It is well known that many recognition modalities often identify the wrong person when position and orientation changes occur in the test set relative to the training set. To combat this problem, a novel statistical approach [45] uses identity surfaces for recognition. With known pose of the head, the face is represented through Kernel Discriminant Analysis (KDA) as a single point. A sparse sampling of the pose space for a given face then allows for the creation of an identity surface. By comparing the partial path of a given series of KDA coded input images to known identity surfaces, better recognition through pose variation is achieved.

In a Support Vector Machine (SVM) hyperplanes are generated in the image representation subspace based on an optimal separation metric. As utilized in [37], an existing algorithm (eigenfaces in this case) is used to reduce the dimensionality of the image space. Hyperplanes in the reduced space are then computed that separate “. . . the largest possible fraction of points of the same class on the same side, while maximizing the distance from either class to the hyperplane” [24].

Other statistical methods include Linear Discriminant Analysis (LDA) [50, 76], kernel methods [49], Bayes classifiers [25], and Adaboost [25, 23]. While each of these methods has advantages and disadvantages, they mostly rely on the quality and quantity of the training data.

Face recognition has also been implemented using several neural network approaches including Dynamic Link Architecture (DLA) [44] and Elastic Bunch Graph Matching (EBGM) [75, 51]. The neural network approach uses dynamic weights to either increase or decrease the importance of specific features used for identification. In this manner, the neural network learns from each face it sees in order to achieve more accurate results in future recognition attempts.

DLA implements a technique referred to as synaptic plasticity, which allows the network to group sets of neurons into symbolic units. Grouping is controlled by a set of connections and temporal correlations between neurons. Connection weights are dynamically adjusted in response to the input (images) resulting in a partitioning of the neural network. Neurons in a given partition then make decisions for only a small part of the input. By convention, grouping via synaptic plasticity is not performed [44].

Within the context of DLA, EBGM uses an elastic graph to recognize individuals. The term “elastic” describes the dynamic adaptation of the graph in size and aspect ratio to better match features of the face [75]. The graph consists of fiducial points (facial landmarks such as the pupils, corners of the mouth, or tip of the nose) and their connections. Feature vectors (sometimes called jets), which represent the same feature from different individuals,

are stored at each graph node. The term “elastic” describes the dynamic adaptation of the graph in size and aspect ratio to better match features of the face [75].

Many other methods have also been considered, including Hyper Basis Function (HyperBF) networks for gender [8], face and voice recognition [9], Radial Basis Functions (RBF) [26, 32], and their generalization (GRBF) as presented in [55].

Feature analysis is widely used for identification by determining a unique measurement from a collection of facial characteristics (i.e., intra-ocular distance, curvature of specific areas of the face, etc.) One early study in feature-based recognition is attributed to Francis Galton [15]. In 1888, he proposed the use of facial profile curves for recognition. These curves are generated by sampling points along the symmetry line of the face. After collecting many facial profile curves, an average curve is computed. The deviation of a person’s profile curve from the average forms the basis for recognition. Much later (in 1964), Woody Bledsoe pioneered the use of computers for recognition using feature points (placed manually) and measures (e.g. Euclidean distance) for recognition [11]. Even though this technique was not automated, it handled significant change in lighting and pose quite well. In [41], a parameter vector is constructed using “internal biocular breadth, external biocular breadth, nose breadth, mouth breadth, bizygomatic breadth, bigonial breadth, distance between lower lip and chin, distance between upper lip and nose and height of lips” [11]. Classification then proceeds as the absolute norm between stored feature vectors and the feature vector obtained from the current target image. In [38] intra-feature distances and angles are used on a database of seventeen male and three female faces. Results show between 45% and 75% correct identification depending on the parameters used.

In other research, the curvature of different regions of the face are used to produce a feature vector [20]. Regions of the face used include the nose bridge (nasion), nose base (base of septum), nose ridge, eye corner cavities (inner and outer), convex center of the eye (eyeball/lid region), eye socket boundary, boundary surrounding nose, and opposing positions on the cheeks for measurement of head width. They report between 70% and 100% accurate identification for 26 subjects.

1.3.2 A Summary of Survey Papers

Many reviews of face recognition are available [3, 11, 14, 60, 73]. Samal and Iyengar (1992) [60] describe several techniques they refer to as nonconnectionist. Most of these techniques operate on 2D images and are concerned with finding intra-feature distances, angles, and areas. A complementary survey by Valentin *et al.* (1994) [73] covers connectionist (statistical) methods of face processing. Connectionist methods of face processing usually take 2D image data and work with pixel values of entire face images (instead of extracting features from a subset of the total pixels for an image as is done in nonconnectionist approaches). Because full images are used in these techniques, the relationships between features within the image, texture, and shape information are preserved. Nonconnectionist and connectionist techniques are also called geometrical and statistical respectively [11, 72, 73]. A different interpretation of “geometrical” is proposed in section 1.3.

Chellappa *et al.* (1995) [11] draw the following conclusions: 1) The upper parts of the face should play a dominant role in recognition, 2) Eigenface and feature point based methods are currently the most developed and should undergo additional testing in realistic situations with thousands of faces, 3) Neural approaches should be developed further and should be tested on much larger databases. The approaches cited in this survey use between 16 and 80 faces.

Barrett (1998) [3] describes solutions to automated face recognition as either neural, eigenface, or wavelet/elastic matching. He then briefly covers some preprocessing steps (e.g., segmentation and normalization) and how some modalities within each method generally work. He concludes that the computational cost of rigid grid Gabor filtering and eigenface approaches are equivalent and that elastic matching, while computationally costly, has superior robustness to facial pose and expression variation.

Zhao *et al.* (2000) [77] present a survey of advances in face recognition techniques since 1995. Of specific interest is their coverage of the FERET and XM2VTS face recognition evaluation procedures. These evaluation procedures allow for the comparison of still and video-based face recognition techniques; a task that has been largely neglected. For details

on the FERET evaluation procedure, see [54, 57, 58, 90]. They conclude that pose and illumination are still unsolved challenges to face recognition and that multi-modal approaches to face recognition show promise.

1.4 Commercial Implementations in Automated Face Tracking and Recognition

Currently there are many commercial face tracking and recognition systems available. For obvious reasons, many companies are reluctant to disclose the technology used in their products. Commercial systems can be grouped into four primary categories: neural network, eigenface, feature analysis, and automatic face processing [94]. Past research in each of these categories, except automatic face processing, is covered in section 1.3.1. Automatic face processing uses distances and ratios of distances between facial features. We list below many commercializations of face recognition technology together with the specific approach used:

- HNeT (Holographic/quantum Neural Technology) Facial Recognition System by AcSys Biometrics Corporation [78]. This neural network approach was developed by John Sutherland. Additional information on this system can be found in [63, 68, 69, 70, 71].
- ZN-Face by ZN Vision Technologies uses an undisclosed neural network approach [101].
- Nvisage by Neurodynamics uses an undisclosed neural network approach [97].
- FaceTools by Viisage uses a proprietary algorithm based on the eigenfaces approach developed at the MIT Media Lab [99].
- Biometrica uses eigenfaces, but does not disclose further details [82].
- FaceIt by Identix (formerly Visionics) uses Local Feature Analysis (LFA) developed by Dr. Joseph J. Atick to generate and measure intra-feature distances for recognition [92].
- Face Guardian by Keyware uses local feature analysis. No information on this product is available on their website [96].

- Visec-FIRE by Berninger Software uses a facial processing approach [81].
- ID2000 by Imagis uses a proprietary wavelet representation of the face for recognition [93].
- BioID uses an undisclosed multimodal system implementing face, voice, and lip movement identification [84, 91].
- FaceVACS by Cognitec applies transforms to specific areas of the face in order to create a user specific feature vector [83].
- UnMask by Vision Sphere Technologies Inc. uses a proprietary feature analysis algorithm [100].
- FaceStation2 by Eyematic uses an unspecified automatic facial expression tracking algorithm to capture and track facial expression for digital character animation [86].
- Face Key by Intelligent Verification Systems uses an undisclosed face and fingerprint recognition algorithm [89].

A comprehensive analysis and comparison of face recognition technologies from AcSys Biometrics, Identix, Viisage and Visionics is available (for a fee) from the International Biometric Group (IBG) [95]. In 2000 and 2002, the Department of Defense Counterdrug Technology Development Program Office (CDTDPO) and other federal agencies cosponsored the Facial Recognition Vendor Test (FRVT) [90] to evaluate several commercial face recognition applications. To the best of our knowledge, other performance comparisons of the major commercial face recognition systems do not exist. Additional information may also be found in [18] on face recognition software offered by Eyematic Interfaces, Identix, Intelligent Verification Systems, Viisage Technologies, and VisionSpheres Technologies.

All these systems work well with specialized training data or in tandem with secondary identification techniques (e.g., finger print, retina scan). However, use in an unconstrained environment is less certain. FRVT 2000 [90] provides definitive performance statistics for

several commercial face recognition systems. It shows that commercial systems can vary from less than 10% to greater than 90% accuracy under different real world constraints (e.g., lighting and distance from the camera). Apparently there is still a need for additional development of algorithms for real world data. While these systems work to some extent, their applicability in the public domain (e.g., in airport or national border security) is questionable. The American Civil Liberties Union (ACLU) offers the following statement on the benefits of face recognition technology being used for security: "...it is abundantly clear that the security benefits of such an approach [automatic face recognition] would be minimal to non-existent, for a very simple reason: the technology doesn't work" [80].

1.5 Conclusions

Most techniques for face tracking and recognition presented in this chapter begin with two dimensional visual spectrum images and are therefore susceptible to variations caused by illumination and imaging processes (as defined in section 1.2.2). Even with the extensive research in this field, illumination is still a key obstacle [77]. Algorithms tested on two dimensional images cannot use the geometric features of the imaged object. The information held in the shape of faces can potentially aid in tracking and recognition. Therefore, techniques that rely on 3D data should be investigated. To that end, we present the methods in chapters 3 and 4. Additional research in the use of shape information for recognition is available [20, 22, 61].

Comparisons between face tracking and recognition systems are very difficult. There are currently two systems (XM2VTS and FERET) used to compare recognition algorithms. However, these methods of comparison have been done for only a few algorithms [77]. A standardized method for quantizing the ability of algorithms to perform face tracking and recognition and the adoption of that standard by the commercial and research communities would be of great benefit.

CHAPTER 2

DATA ACQUISITION AND REPRESENTATION

We are interested in utilizing three dimensional shapes of faces for recognition. Hardware for capturing shape information is increasingly common. Therefore, techniques that use shape information deserve careful study. We represent facial shape as a mesh. Meshes are acquired using a Minolta Vivid 700 3D camera capable of capturing both geometry and texture. The measuring principle of the camera is illustrated in figure 2.1 [52]. By passing a laser light through a cylindrical lense, a horizontal light-stripe is created (illustrated as a dotted line coinciding with the axis of rotation of the galvano mirror in figure 2.1). The light-stripe is then reflected by a galvano mirror onto objects in the imaged scene. A CCD (charge-coupled device) then receives the light-stripe reflected by the scene and generates distance information by triangulation. The galvano mirror is then rotated resulting in a projection of the light-stripe onto a different part of the scene and the measuring begins again. This is done 200 times for one range scan. The geometry resolution for the camera is 200×200 or, 40,000 points. The texture resolution is 400×400 or, 160,000 pixels. The resolution achieved when capturing a face is approximately 10,000 points (15,000 triangles) and 40,000 pixels. The resolution of the data captured for a given face depends on its size, shape, and position relative to the camera. The face geometry captured by the Vivid 700 does not depend on the illumination of the imaged scene. However, the texture capture mechanism is sensitive to illumination as is any visible spectrum imaging technique. To minimize this, the same fixed lighting is used for all our data captures.

A two dimensional mesh is a discrete approximation of a 2D surface in \mathfrak{R}^3 (see figure 2.2). It is defined by a set of sampled points on a surface and a connectivity map. The sampling is done by the Minolta Vivid 700 3D camera and the connectivity map is generated

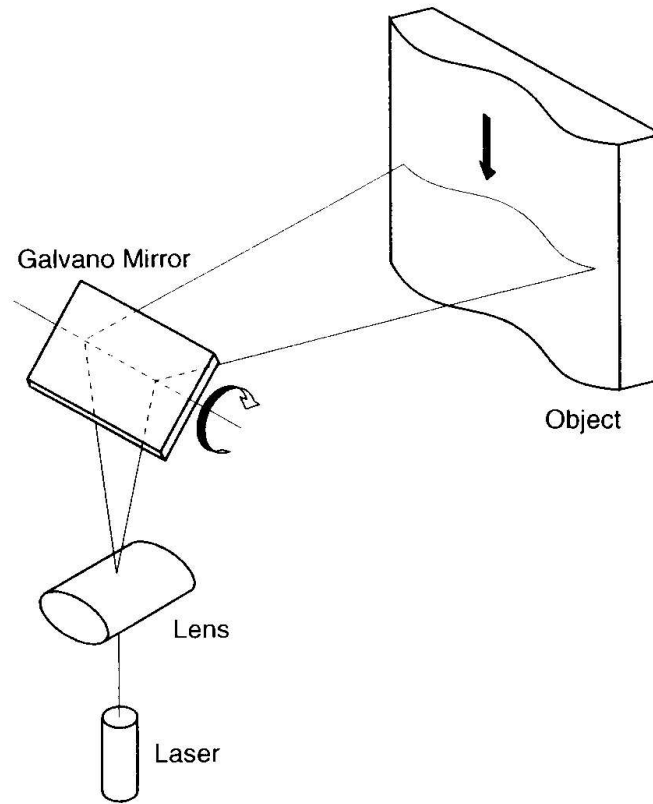


Figure 2.1. An illustration [52] of the measuring principle used by the Minolta Vivid 700 camera. By passing a laser light through a cylindrical lens, a horizontal light-stripe is created. The light-stripe is then reflected by a galvano mirror onto objects in the imaged scene. A CCD then receives the reflected light-stripe and generates distance information by triangulation. The galvano mirror is then rotated resulting in a projection of the light-stripe onto a different part of the scene and the measuring begins again. This is done 200 times for one range scan.

automatically by the Minolta software accompanying the camera. Two typical meshes are displayed in figure 2.2. Meshes are saved in the Alias Wavefront (OBJ) file format using the Minolta software.

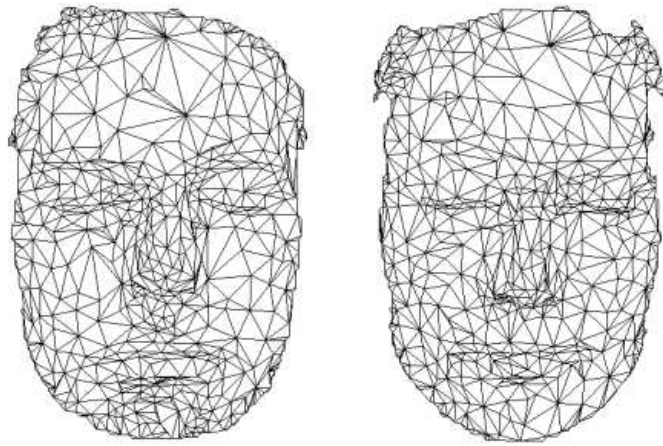


Figure 2.2. Facial meshes captured using the Minolta Vivid 700 3D camera. These meshes have been decimated for illustration purposes to 1,000 triangles each down from a typical 15,000 triangles.

CHAPTER 3

FACE TRACKING USING IMAGES GENERATED FROM GEOMETRY

Video is everywhere. Common uses of video include surveillance in public areas of known persons or suspicious activity, security systems, sporting events, personal pleasure, etc. In many of these instances, the goal of video recording is to track or identify one or more people or objects captured in the video. Tracking can be achieved through body motion characteristics and many parts of the body have been used [5, 16, 59, 43, 46, 28, 61]. We consider specifically tracking of face position and orientation through a 3D space from their projection onto a 2D image. We refer to this as face tracking.

In this chapter we present an algorithm for the generation of conditional mean estimates of position and orientation of a face in a video. Data representation and equipment used for the experiment were explained in chapter 2. The algorithm is detailed in section 3.1, experiments and results are explained in section 3.2, followed by conclusions and future work.

3.1 Algorithm

Given a video of an object translating and rotating in space, our goal is to reconstruct its transformation matrix as a function of time. A transformation matrix is a precise mathematical description of the position and orientation of an object. It is unique for each frame of a video. An illustration of our basic technique is given in figure 3.1. We use two objects to demonstrate the algorithm: a synthetic textured box and a face (figure 3.3); both objects are three-dimensional. To estimate the transformation matrix of the face in the video we render the corresponding mesh under an estimated transformation matrix onto

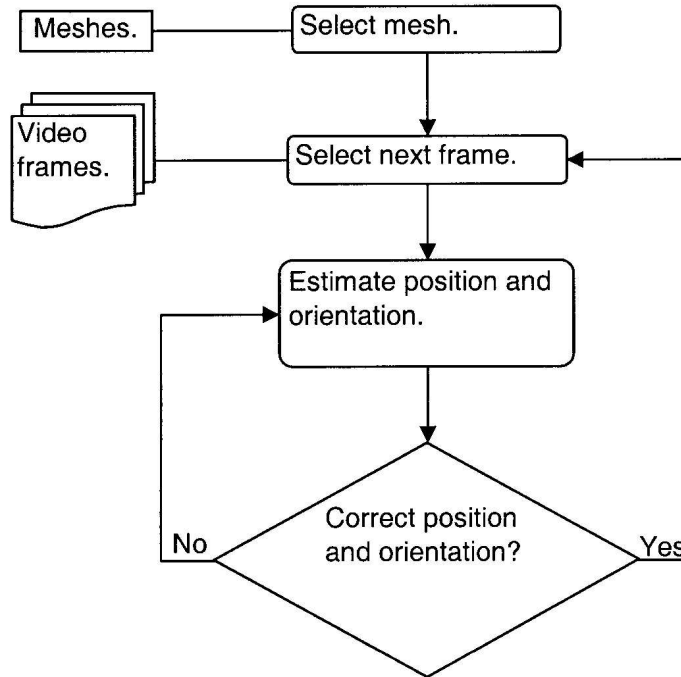


Figure 3.1. An illustration of the basic flow of the tracking algorithm. The mesh corresponding to the object to track is manually chosen.

a 2D plane and compare that rendering to a given frame of the video. For each frame, the error is measured as the L2 norm between the video frame and the rendered image. We use scaled simple iterations (a descent method) ([65], pp. 47) to minimize the magnitude of the error. Translational and rotational velocities are computed to aid initialization. Once the error is minimized, the process is repeated for the next video frame using the current estimated position and orientation updated by the computed velocities as an initial guess.

In what follows, positions and orientations in \mathbb{R}^3 are indicated by $\mathbf{x}=(x_1, x_2, x_3)$ and $\boldsymbol{\theta}=(\theta_1, \theta_2, \theta_3)$ respectively. The video is composed of a set of frames \mathbf{T}_k , indexed by k . We have no true knowledge of the position (\mathbf{x}) or orientation ($\boldsymbol{\theta}$) of the object in the first frame of the video (\mathbf{T}_1). We do not address segmentation and recognition (two active fields of research in the area of automated tracking). Instead we assume that the face we wish to

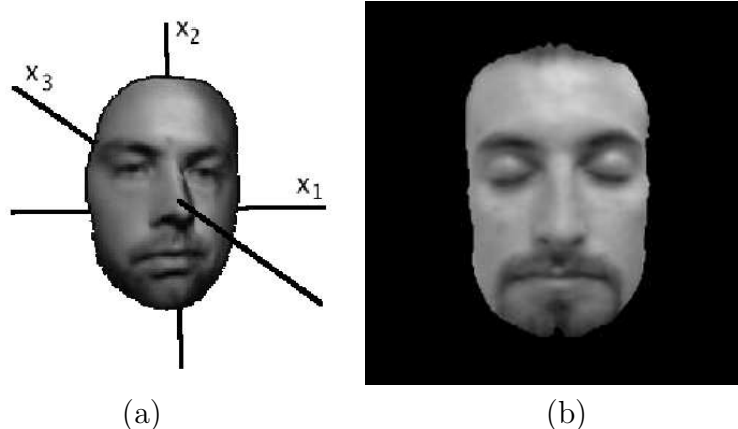


Figure 3.2. (a) A rendered image of a face displayed over the rendering axis. (b) A rendering of the assumed position, orientation, and scale of a face in the first frame of the video.

track is known, approximately centered, and face forward in the first video frame. Figure 3.2 presents a centered, face forward rendered face image over the rendering axis (the axis and image are slightly rotated so that the x_3 axis does not appear as a point). This is the assumed position, orientation, and scale of a face in the first frame of the video (i.e., video is chosen so that these assumptions are valid). Rendering parameters (\mathbf{x} and $\boldsymbol{\theta}$) are then set to mimic these assumptions during estimation for the first frame. Since we do not address identification in this algorithm, we manually choose the mesh corresponding to the subject in the video. Let $\mathbf{E}(\mathbf{x}, \boldsymbol{\theta})$ be the projection of the 3D object onto the image plane subject to translation \mathbf{x} and rotation $\boldsymbol{\theta}$. An image is defined as a 2D rectangular grid $I_{m,n}$. A distance measure (cost function) between the target image (\mathbf{T}_1) and estimate image ($\mathbf{E}(\mathbf{x}, \boldsymbol{\theta})$) is defined by

$$C(\mathbf{x}, \boldsymbol{\theta}) = \|\mathbf{T}_1 - \mathbf{E}(\mathbf{x}, \boldsymbol{\theta})\|_2, \quad (3.1)$$

where the L2 norm of an image I is defined as

$$\|I\|_2 = \sqrt{\sum_m \sum_n I_{m,n}^2}. \quad (3.2)$$

Let $\mathbf{Q} = (\mathbf{x}, \boldsymbol{\theta}) = \{q_i\}$ be the six dimensional vector of unknown parameters. We seek to minimize $C(\mathbf{Q})$ over all possible values of \mathbf{Q} . This is achieved when

$$\frac{\partial}{\partial q_i} C(\mathbf{Q}) = (-2) \left\langle \mathbf{T}_1 - \mathbf{E}(\mathbf{Q}), \frac{\partial \mathbf{E}(\mathbf{Q})}{\partial q_i} \right\rangle = 0 \quad \text{for } i = 1 \dots 6. \quad (3.3)$$

(A derivation of equation 3.3 can be found in Appendix A.) Due to the nonlinear nature of equation 3.3, we apply a descent method to solve for \mathbf{Q} :

$$\begin{aligned} \hat{q}_i &= q_i + \delta \frac{\partial}{\partial q_i} C(\mathbf{Q}) \\ \hat{\mathbf{Q}} &= (q_1, \dots, \hat{q}_i, \dots, q_6). \end{aligned} \quad (3.4)$$

If $C(\hat{\mathbf{Q}}) < C(\mathbf{Q})$ then the estimated position is kept ($\mathbf{Q} = \hat{\mathbf{Q}}$), otherwise it is rejected. A constant scale factor, $\delta > 0$, is chosen empirically for each object. If δ is too large the updated parameter values (\mathbf{Q}) will oscillate without $\frac{\partial C}{\partial q_i}$ converging to zero. If δ is too small the convergence will be slow. The algorithm continuously cycles through i until $|C(Q)|$ falls below a specified tolerance β , chosen empirically. The value of β depends on noise in the video and the time you're willing to wait for convergence.

Once $C(\mathbf{x}, \boldsymbol{\theta}) < \beta$ for T_1 , the algorithm concludes that the position and orientation of the mesh in the estimated image is correct. The estimated position and orientation are then stored, and the estimation process moves to the next target image. For the second target image, the estimated position and orientation of the mesh for frame \mathbf{T}_1 are used as the initial condition. Otherwise, the same estimation steps as previously explained are used. When the position and orientation estimation for the second frame is correct, the estimates are stored again.

Now that the position and orientation of the object in the first two frames has been successfully estimated, position and orientation velocities are computed. The position (translation) velocity estimate ($v_{\mathbf{x}}$) is computed as the difference in the estimated position for the previous two frames ($T_k - T_{k-1}$). The orientation (rotation) velocity estimate ($v_{\boldsymbol{\theta}}$) is computed in the same manner using the estimated orientation for the previous two frames. The velocities $v_{\mathbf{x}}$ and $v_{\boldsymbol{\theta}}$ are then added to \mathbf{x} and $\boldsymbol{\theta}$ found for \mathbf{T}_2 and used as the initial condition ($\mathbf{E}(\mathbf{x} + v_{\mathbf{x}}, \boldsymbol{\theta} + v_{\boldsymbol{\theta}})$) on the next target (\mathbf{T}_3). This is done to reduce the time needed to correctly estimate the position and orientation for sequential frames. Position ($v_{\mathbf{x}}$) and

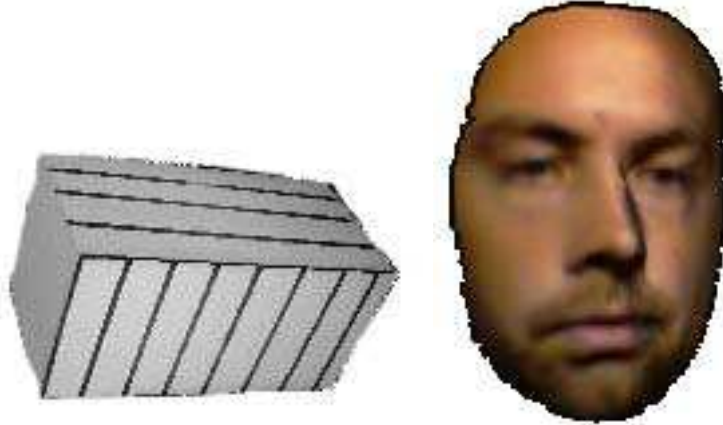


Figure 3.3. Pictures of a synthetically generated box and a scanned face. Textures for the box were acquired using a digital camera and mapped to a hand generated box manually. The mesh and texture of the face was acquired automatically using a Minolta Vivid 700 range camera.

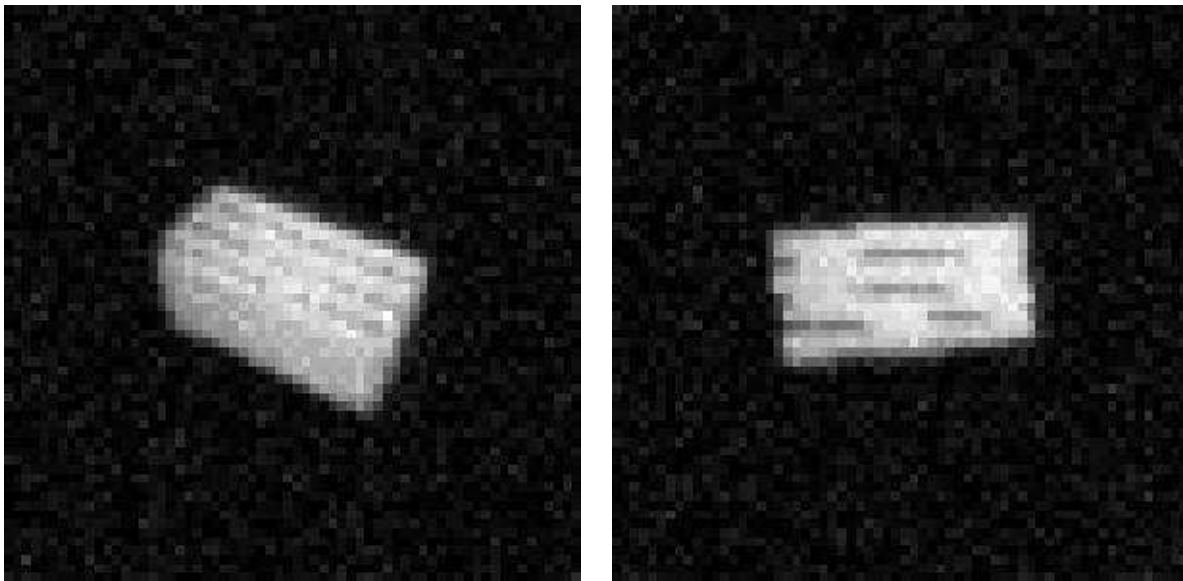
orientation (v_{θ}) velocities are now updated after each correct target estimate ($C(\mathbf{x}, \boldsymbol{\theta}) < \beta$) using the position and orientation estimates for the previous two frames.

3.2 Experiment and Results

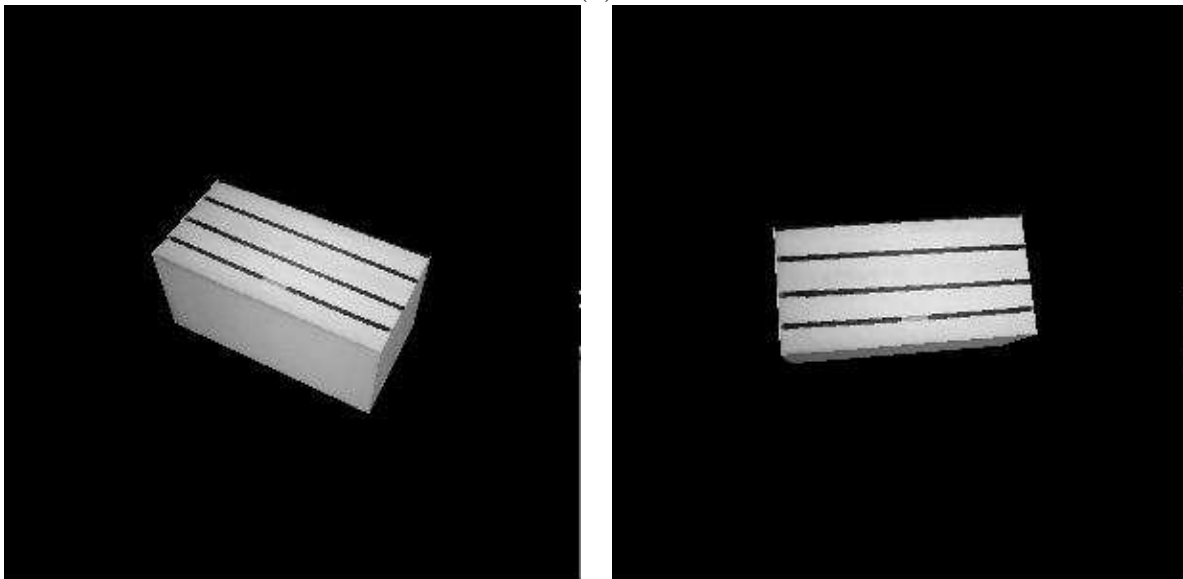
The first step in implementing this algorithm is to approximate $\frac{\partial \mathbf{E}}{\partial q_i}$ in equation 3.3 by

$$\frac{\partial \mathbf{E}}{\partial q_i} \simeq \frac{\mathbf{E}(q_1, q_2, \dots, q_i + \Delta, \dots, q_6) - \mathbf{E}(q_1, q_2, \dots, q_i, \dots, q_6)}{\Delta_i}. \quad (3.5)$$

The input video for tracking can be either synthetically generated using a textured mesh and position/orientation equation or captured using standard video equipment. In this set of experiments we use two videos of 3D objects under simultaneous translation and rotation. For testing purposes we initially use a simple box with textures as shown on the left of figure 3.3 in the generation of a synthetic video. Use of a synthetically generated target video allows us to work with an object with a minimum of unknowns that could influence the result (e.g., distortion and noise induced by the video capturing system). The OpenGL and GLUT rendering libraries were used for rendering under the perspective projection (figure 3.5).



(a)



(b)

Figure 3.4. (a) Two synthetic target images with Gaussian noise. Directly below each target image, part (b) shows the corresponding best estimates to the target's position and orientation.

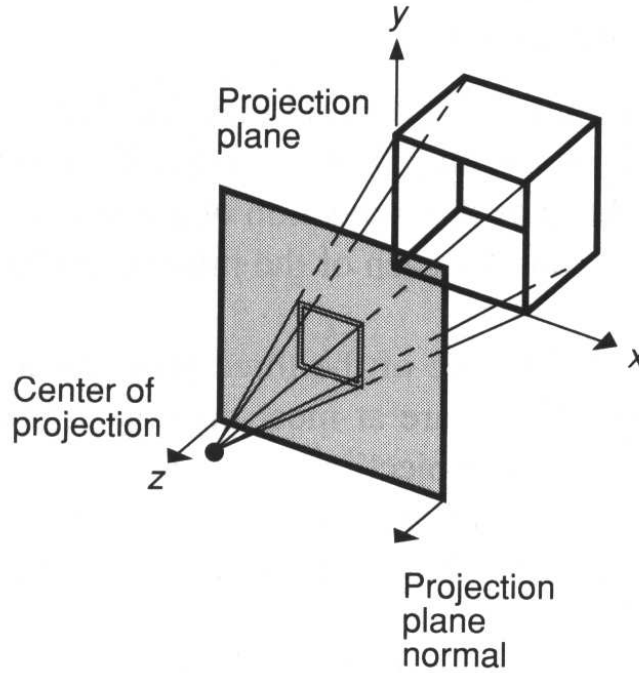


Figure 3.5. An illustration of the perspective projection concept [13]. This projection was used in the generation of the synthetic target tracking video for the box.

Figure 3.4 displays two targets from the synthetically generated video and their associated best estimate images. To simulate the noise associated with real video capture equipment, Gaussian noise ($N(0, 0.5)$) is added to each target image. For this experiment we chose the following parameter values: $\Delta_{\mathbf{x}} = 0.001$ in equation 3.5; $\Delta_{\theta} = 1.5$; $\delta_{\mathbf{x}} = 0.001$ in equation 3.4; $\delta_{\theta} = 0.001$; $\beta = 50$. Parameters $\Delta_{\mathbf{x}}$, Δ_{θ} , $\delta_{\mathbf{x}}$, δ_{θ} , and β are chosen empirically through observation during several trial runs. In figure 3.6 two frame of a video captured using a Sony XC-003 digital frame grabber and the estimated positions/orientations are presented. The choice of an appropriate step size in equation 3.5 for the box and the face was challenging. Many choices of Δ_i caused the sign of the partial derivative to change. This can result in the cost of the new estimate exceeding the cost of the previous estimate. When this happened for every q_i in \mathbf{Q} , the algorithm would fail without finding an estimate for the

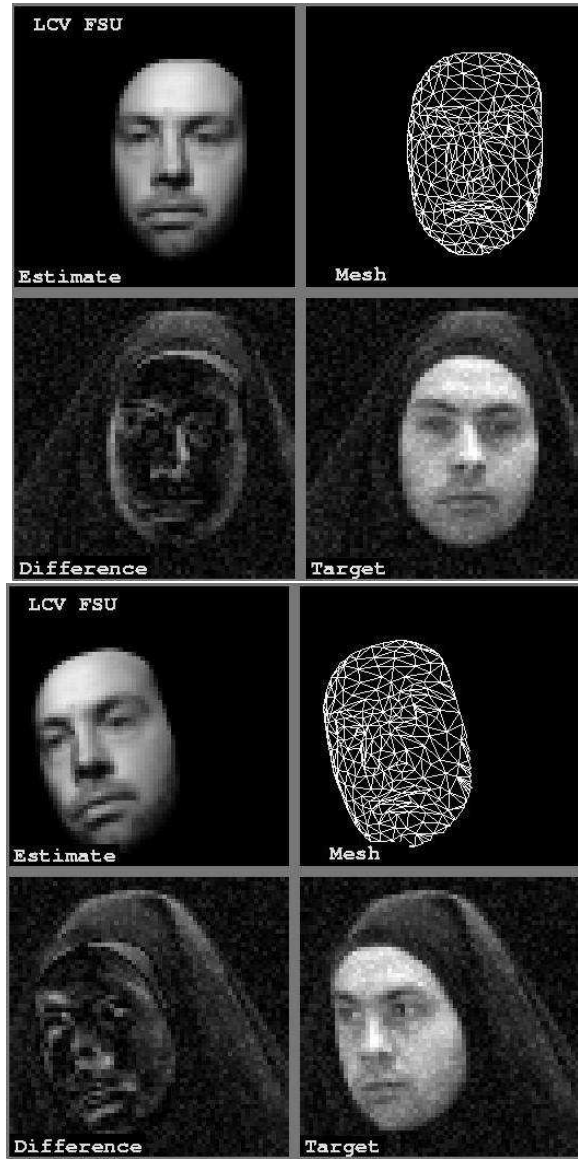


Figure 3.6. Two target frames from a video captured using a Sony XC-003 digital frame grabber are shown (labeled “Target”). The apparent shroud is a black shirt used to lessen background clutter. Gaussian noise ($N(0,0.5)$) was added to the targets to simulate poor video clarity. The image labeled as “Estimate” shows the corresponding face mesh rendered at a position and orientation where $C(\mathbf{x}, \boldsymbol{\theta}) < \beta$ (see section 3.1). The image labeled “Mesh” shows the textureless mesh (at a reduced resolution) used for the conditional estimates shown in the image labeled “Estimate”. The image labeled “Difference” is a visualization of $\mathbf{T}_k - \mathbf{E}(\mathbf{x}, \boldsymbol{\theta})$ for the estimate shown above it.

position and orientation of the face in the target video frame. After many (about 10) trial runs, the same values for Δ , δ , and β were chosen for the box and the face.

An SGI Octane running at 195Mhz with 768Mb of RAM was used to conduct these experiments. Tracking the first two frames took 30 to 40 seconds (due to the lack of velocity estimates as described in section 3.1). Tracking additional frames took 5 to 10 seconds per frame.

3.3 Conclusions and Future Work

The technique outlined above can track a face undergoing smooth changes in position and orientation relative to the video frame rate (the velocities v_x and v_θ change slowly). However δ and β must be chosen carefully. Several (on the order of 10) trial runs of the tracking algorithm with various values for these two variables are needed for empirical estimation. About 20 to 50 frames can be tracked before the algorithm fails. Since the initial two frames do not use any prior knowledge of the target scene, their computation time is noticeably longer than most other target frames. Estimation on the first two frames converges in about twenty iterations while convergence in subsequent frames takes approximately ten iterations. In its current form, this algorithm is not a practical solution for face tracking in unconstrained environments. This technique does not handle backgrounds, which can easily lead to very inaccurate estimates. This is due to the cost function shown in equation 3.1, which has no knowledge of facial shape. Some knowledge of face shape, skin tones, feature distances, background subtraction, or face extraction could be used to aid in this deficiency by limiting the size of the face relative to the image, or eliminating the background completely.

CHAPTER 4

FACE RECOGNITION USING RANGE IMAGES

The problem of recognizing people from their facial images has gained wide attention in recent times. This problem has been studied using several sensing modalities, such as visible spectrum and infrared, and using several pattern recognition techniques. For visible spectrum imaging, there have been many studies reported in the literature. Some published techniques include feature point extraction on the face by Bledsoe [7], feature point extraction on the face and body parts by Kelly [42], segmentation, geometrical parameterization of feature points by Kanade [38], eigenfaces by Pentland [72], neural networks by Golomb [17], and profiles by Kaufman [40]. Methods based on visible spectrum images are limited in their applications for several reasons. The main reason comes from the variability generated in images due to variation in illumination sources. The images change drastically with changes in the location and intensity of the illumination. Research in face recognition has shown that the variation in facial imaging due to illumination is greater than the variation due to identity [12, 18].

In general there is great interest in the development of mathematical models that capture the variability among face images [27]. A long-term strategy is to model the physical factors that lead to differences among face images. These factors are: (i) shapes of facial surfaces, (ii) facial textures, (iii) illumination models, and (iv) modeling of pose relative to the camera.

A possible solution is to involve other sensors that are not sensitive to visible light. Toward that goal, infrared cameras have been used in face recognition [66]. However, we focus on another imaging modality: range imaging. In range imaging, the image captures the distance (range) of the nearest object along each pixel. Although the range scanner is an optical imaging device, its wavelength of operation is robust to changes in illumination.

Also, range images measure the physical shape of the surface and ignore the color (texture), and therefore are robust to texture variations.

We restrict ourselves to the development of mathematical models for only the variations of facial surfaces in \mathfrak{R}^3 . The surfaces are captured using a high-resolution 3D scanner and represented by a mesh as described in chapter 2. However, the dimensionality of the captured data makes direct computation using the mesh difficult. For example, a mesh representation of a facial surface containing 10,000 vertices is an element of 30K dimensional space. Ideally, one would like to compare the facial surfaces in \mathfrak{R}^3 to perform recognition, but very few metrics for surfaces in \mathfrak{R}^3 are known [27, 79]. Since it is difficult to analyze variability or impose probability models on this space, the first task is to reduce the dimensionality while retaining most of the variability in the observed surfaces. Our approach is to map the shape information encoded in a mesh to the image plane (a two dimensional image) and use conventional image analysis techniques for recognition. To further reduce data dimensionality, we propose to study three projection methods that reduce two dimensional images to a low-dimensional Euclidean space, namely, principal component analysis (PCA), independent component analysis (ICA), and Fisher's discriminate analysis (FDA). Having obtained low-dimensional representations of the observed surfaces, the next task will be to develop a classification algorithm under a certain choice of metrics.

In the following sections we describe the process of data collection (section 4.2), the generation of range images (section 4.3), and the registration of these images using certain feature points (section 4.4). In section 4.6 we use PCA, ICA, and FDA to reduce image dimensionality followed by an explanation of the identification process (section 4.7) and experimental results (section 4.8). Finally, conclusions and future work are discussed.

4.1 Representation of Facial Surfaces

We are interested in mathematically representing and analyzing shapes of facial surfaces. For simplicity of analysis, we have chosen to use range images of faces to represent the facial shape variability. Since the 3D scanner used in collecting facial surfaces provides data in



Figure 4.1. An example data capture session. Subjects stay in a predetermined position and orientation.

the form of meshes, one needs to preprocess this data into range images before image based recognition techniques can be applied. We start by describing the data acquisition process.

4.2 Data Acquisition

The data acquisition presented in this section follows the technique presented in chapter 2. We have utilized six face meshes per person, each with a unique expression. To acquire each mesh, subjects are asked to stay in a predetermined position and orientation with respect to the camera (figure 4.1) resulting in a rough, global registration of meshes. Subjects are then asked to make six different facial expressions (neutral, smile, frown, angry, squint, and scared) and each expression is captured. Next we describe the task of converting a triangulated mesh into a range image.

4.3 Generation of Range Images

A range image (of a mesh) is an array of depth values from each triangle of the mesh onto a 2D image plane with the image plane being perpendicular to the camera view. Depth values are immediately known for triangle vertices. However, the vertices of a given triangle may not project uniformly to adjacent pixels, leaving holes in the range image. Therefore we must find the pixels in the image plane that are interior to the projection of a given triangle (i.e., if we draw lines connecting the projections of the vertices on the image plane to form a triangle, interior pixels lie within the triangle.) To determine which pixels in the image plane lie inside the projection of a given triangle we utilize the Line Crossing Algorithm [1]. First we define a limited search space within the image plane; then, using the Line Crossing Algorithm, we test each pixel in that space for inclusion within the projection of the triangle on the image plane. The following description illustrates our implementation of the Line Crossing Algorithm.

Let the three vertices of a triangle in a mesh be $P1$, $P2$, and $P3$ as shown in figure 4.2. For a given vertex, P_k , let the coordinates of that vertex be given by (x_k, y_k, z_k) . For that triangle define a quadrilateral on the image plane by the maximum and minimum x and y vertex values, $(xMax, yMax)$, $(xMax, yMin)$, $(xMin, yMin)$, $(xMin, yMax)$:

$$\begin{aligned} xMax &= \max(x_1, x_2, x_3) & xMin &= \min(x_1, x_2, x_3) \\ yMax &= \max(y_1, y_2, y_3) & yMin &= \min(y_1, y_2, y_3) \end{aligned} \tag{4.1}$$

The pixels inside the quadrilateral are then individually tested (using the Line Crossing Algorithm) to determine if they are interior or exterior to the projection of the triangle on the image plane. To implement the Line Crossing Algorithm, first project the vertices $P1$, $P2$, and $P3$ onto the image plane by discarding the z coordinate of each vertex. After selecting a pixel, P , within the quadrilateral described above, project $P1$, $P2$, $P3$, and P onto the y -axis of the image plane. If the projection of P lies between the projections of the end points of any two edges, then P may be interior; otherwise P is exterior. Using the y value of P , solve for x on the two edges found in the previous step and project P and those

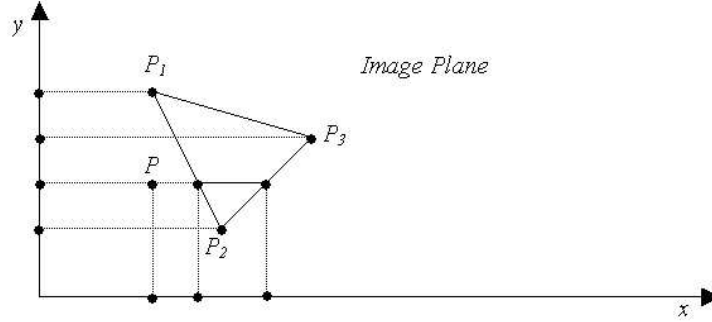


Figure 4.2. An illustration of the Line Crossing Algorithm. The point in question, P , is outside the triangle (exterior). This graphic is borrowed from [1].

x values onto the x -axis. If the x component of P lies between these two projections, P is interior; otherwise it is exterior. In figure 4.2, P is within the projection of the end points for two edges of the triangle onto the y -axis. However, it is not within the projection of those end points onto the x -axis. Consequently, P is exterior to the triangle.

For each interior pixel on the image plane, we compute the height (the z value) of that pixel from the mesh. To do this, the vertices of a triangle ($P1$, $P2$, and $P3$) are used to generate the equation of a plane. Organize $P1$, $P2$, and $P3$ as a matrix,

$$A = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix}. \quad (4.2)$$

Then check for collinearity by computing the determinant, $\det(A)$. If the determinant of A is non-zero then the three points are not collinear, and we solve for the coefficients to the equation of a plane:

$$[c_1 \quad c_2 \quad c_3] = [z_1 \quad z_2 \quad z_3] \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1}. \quad (4.3)$$

(If $\det(A)$ is zero, the degenerate triangle is skipped.) The height of each interior pixel P is then found using

$$z = c_1 P_x + c_2 P_y + c_3. \quad (4.4)$$

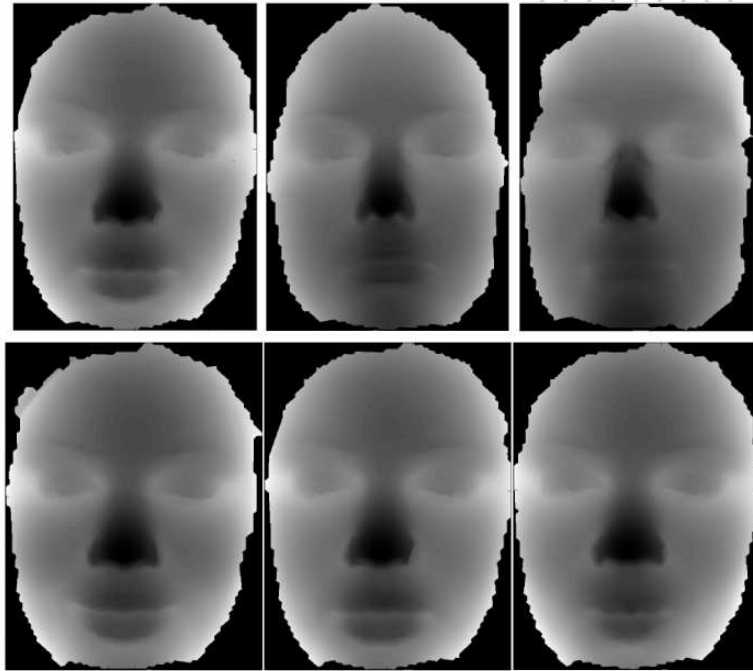


Figure 4.3. Facial range images. The top row of images displays three, unique persons with the same expression. The bottom row of images displays the same person under three different facial expressions. Aside from the background, light areas of range images indicate portions of the face further from the range camera while darker areas indicate portions closer to the range camera.

In this way, each triangle in the mesh is traversed and orthographically projected onto the image plane. Note that differently scaled meshes lead to different range images.

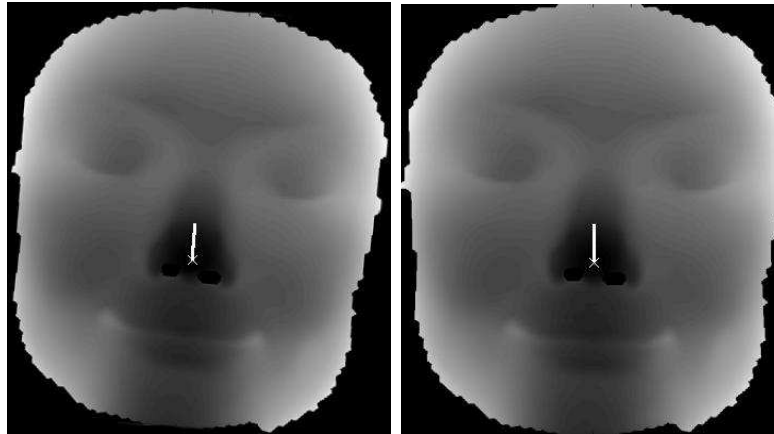
Shown in figure 4.3 are six example range images generated using this technique. The top row of images displays three different people imaged at the same expression. The bottom row of images displays the same person under three different facial expressions. In these range images, lighter pixels indicate parts of the face further from the camera while darker pixel values indicate parts of the face closer to the camera, and completely black pixels indicate no data. Note that the central portion of each range image corresponding to the nose is not black, but dark gray, meaning that the nose is the closest part of the face to the camera.

4.4 Planar Registration of Range Images

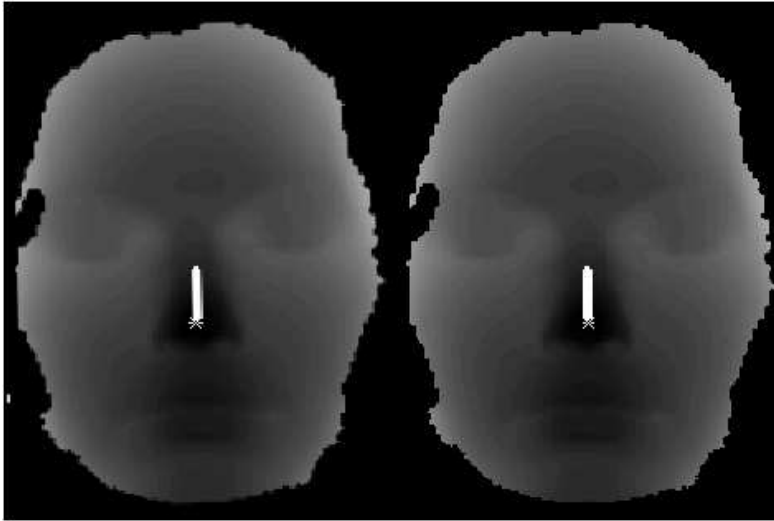
There are two sources of undesired variability in range images: orientation and position of the subject’s face in \mathfrak{R}^3 relative to the camera during data capture. This is first dealt with by controlling the data capture environment as described in section 4.2. To account for some of the remaining variability, correctional rotation, translation, and depth adjustments are made to the range images; each step improves the registration between range images through feature alignment. All registration is performed on the 2D range images, avoiding the computational complexity associated with 3D registration. We have chosen to automatically align images using two salient features: nose tip and the bridge of the nose.

We desire the bridge of the nose to be vertical in the observed faces. Since this is not always true, a rotational correction is necessary. Rotational registration reduces the error induced by small rotations of the subject’s face in the image plane. In figure 4.4 we see range images for two subjects before (left) and after (right) rotational correction. Since each facial mesh was captured with the subject directly facing the camera (face forward), we assume the tip of the nose to be the part of the face closest to the camera. We say the pixel corresponding to the tip of the nose in the range image has the “strongest” intensity. We call this the tip pixel. (If this is not true for a given subject, then that subject is not used in our experiment.) The tip pixel is located by searching over every pixel value in a given range image. After finding the tip pixel (indicated by an asterisk in figure 4.4) we inspect several successive rows above the row containing the tip pixel. Along each row we choose the pixel with the strongest intensity (relative only to that row). This procedure leads to a number of two-dimensional points appearing as a line along the bridge of the nose (left images in figure 4.4.) These points are fed to a line-fitting algorithm that returns an angle. This angle is the rotation necessary to make the line, and therefore the face, vertical.

Next, each range image is translated in the image plane so that the tip pixel (indicated by the asterisk in 4.4) corresponds to the center point location. This enhances the positional alignment of the subject’s face in the image plane.



(a)



(b)

Figure 4.4. In these range images, the asterisk indicates the pixel of *strongest* intensity. The line extending from the asterisk toward the forehead indicates the pixel of strongest intensity for a number of rows. The image pairs illustrate range images before rotational correction (left) and after (right).

Finally, depth adjustment restricts variability in the subject's distance from the camera during data collection. A constant is chosen as the tip pixel value for all range images. In other words, a constant is added to every pixel in the range image so that the tip pixel has the same value for all range images.

Through this process we obtain range images (roughly 659×500) of each subject under each facial expression. We have now moved from a 30K dimensional space to more than 300K dimensional space. This initially appears to be a step in the wrong direction. However, in the next three sections we will see how standard image analysis techniques can be used on these images for efficient recognition.

4.5 Preprocessing of Range Images

In the previous section, meshes (elements of \mathbb{R}^3) were projected into range images (elements of \mathbb{R}^2). Through this projection, data moved from a 30K dimensional space to a 300K dimensional space. However, this step was necessary so that standard image analysis techniques could be used. A series of three preprocessing steps is also required: resizing, masking, and patching.

Range images are resized so that later computations are tractable. Unprocessed range images are elements of high dimensional space (roughly 659×500). They are resized to 10% and 60% of the original size resulting in image of roughly 70×50 and 300×200 pixels. This reduction in resolution is necessary due to limited computational resources. Next each range image is masked.

The black and white mask shown in figure 4.5 is used to crop each range image identically by performing a corresponding pixel comparison; only areas within the black border are kept.

Next, holes in each range image are patched. A hole is any pixel in the range image that lies inside the mask (figure 4.5) but does not have an intensity value (its value is zero, which represents no data.) A hole at pixel $P_{i,j}$ for a given range image is patched by averaging adjacent pixel values:

$$P_{i,j} = \frac{P_{i-1,j-1} + P_{i-1,j} + P_{i-1,j+1} + P_{i,j-1} + P_{i,j+1} + P_{i+1,j-1} + P_{i+1,j} + P_{i+1,j+1}}{8}. \quad (4.5)$$

Only adjacent pixels with non-zero values are used in equation 4.5 (i.e., missing data does not contribute to the value of pixel $P_{i,j}$). Shown in figure 4.6 are two range images. A black dot represents a hole in the image on the left, while the image on the right has been patched.

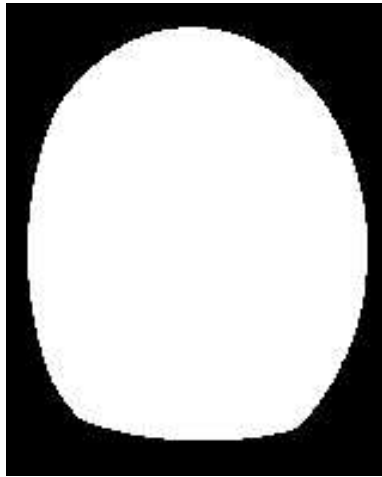


Figure 4.5. This mask is used to crop range images so that every range image has data in corresponding pixels. This is the second of three preprocessing steps before the use of dimension reduction techniques.

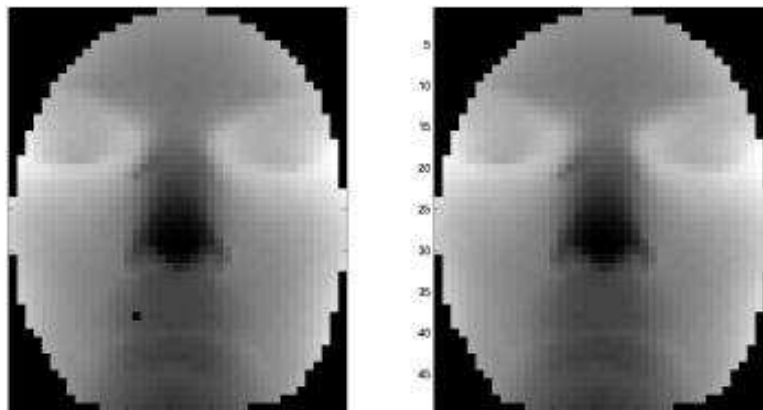


Figure 4.6. Range images generated from a facial mesh. The black dot in the left image represents a hole in the range image. The same range image is displayed on the right after hole patching. Lighter pixel values indicate parts of the face further from the camera while darker pixel values indicate parts of the face closer to the camera. Completely black areas indicate no data. Note that the nose is dark gray, but not black. This is the final preprocessing step before the use of dimension reduction techniques.

4.6 Dimension Reduction

There are many different reduction techniques available including Independent Component Analysis (ICA) [33, 34], Linear Discriminate Analysis (LDA) [2], Principal Component Analysis (PCA) [36], and Fisher’s Discriminate Analysis (FDA) [47]. In this paper we have chosen to focus on PCA, ICA, and FDA. Each reduction technique allows us to generate an orthogonal basis for a low dimensional subspace in the high dimensional space of range images.

Because each reduction technique is sensitive to noise, the preprocessing steps described in section 4.5 are implemented.

For PCA we use the "svd" function in Matlab. The computed basis for PCA allows for the reconstruction of faces from the basis with minimal error in the least squares sense. To accomplish this, the first basis lies in the direction of greatest variation in the data set; the next basis lies in the direction of greatest variation in the data set perpendicular to all previously computed basis vectors. Since PCA does not use any information about which class a given face belongs to, it may well be the case that the basis with the best distinguishing ability is not used in the PCA reduction.

To perform reduction the data is first reorganized. We reshape each of the k range images (R_0, R_1, \dots, R_k) into vectors. Let v_i be the vector obtained by reshaping the range image $R_i \in \mathfrak{R}^{m \times n}$:

$$v_i = [R_i(0, 0), R_i(0, 1), \dots, R_i(m, n)]^T \in \mathfrak{R}^{mn \times 1}. \quad (4.6)$$

Then place image vectors vertically into a matrix A where each column in A is one range image:

$$A = [v_1, v_2, \dots, v_k] \in \mathfrak{R}^{mn \times k}. \quad (4.7)$$

Reduction is performed on A yielding an orthogonal basis, $X \in \mathfrak{R}^{mn \times k}$, in the space of k range images. To reduce the dimensionality of a range image I we reshape it into a column vector (equation 4.6) and project it into the orthogonal basis X using the inner product,

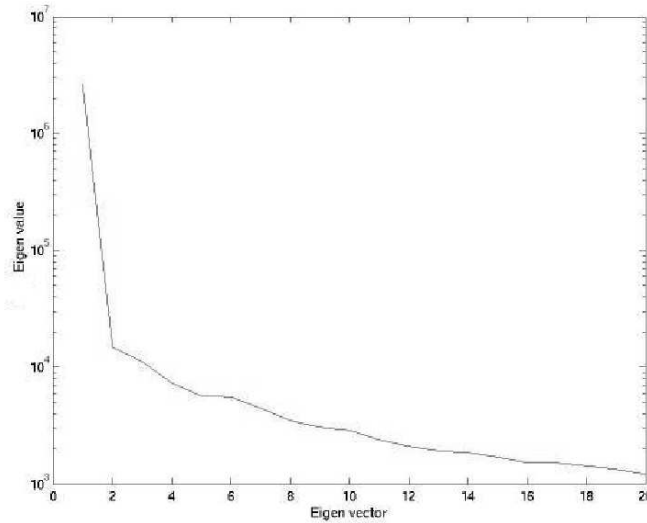


Figure 4.7. The first 20 eigenvalues plotted along the horizontal axis in order of decreasing value.

$$\begin{aligned}
 I^T \cdot X &= P \text{ where } I^T \cdot X_1 = P_1, \\
 I^T \cdot X_2 &= P_2, \dots I^T \cdot X_{10} = P_{10}.
 \end{aligned}
 \tag{4.8}$$

Because we use a subset of the basis, this projection P results in a concise description of the range images in the training set in terms of the computed space. For PCA we use the eigenvectors associated with the 10, 30 and 60 largest eigenvalues so reduced representations of range images are 10, 30, and 60 dimensional.

For ICA we use the FastICA [35] implementation from Helsinki University of Technology which utilizes Hyvarinen’s fixed point algorithm. ICA finds basis vectors (a.k.a., independent components) that are as independent as possible. To use this method, we assume the observed signals (faces in our case) do not fit a gaussian distribution. One can think of the independent components found using ICA as the most basic building blocks for the observed signals. Like PCA, ICA knows nothing about the class that signals originate from making it suboptimal for discrimination.

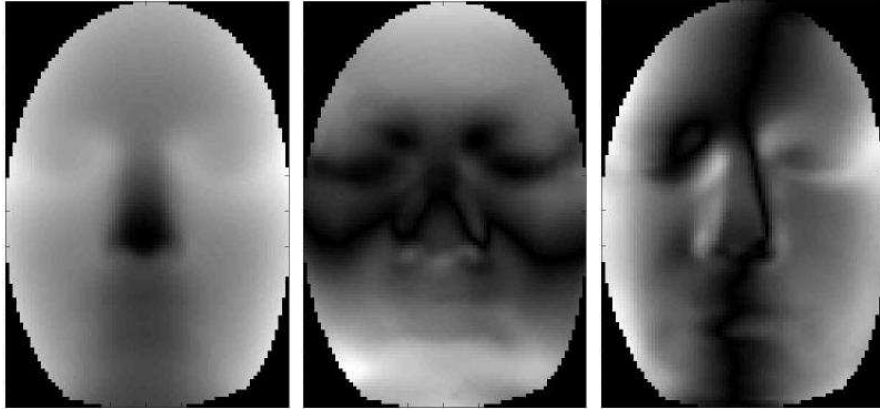


Figure 4.8. Three eigenvectors associated with the three largest eigenvalues in descending order from left to right. Lighter pixel values indicate regions of greater variation and darker pixel values indicate regions of less variation.

For FDA we use software developed by Dr. Xiuwen Liu (Department of Computer Science, Florida State University) based on [4]. FDA is one type of linear discriminate (i.e., class discrimination is at the heart of FDA). By minimizing within class variance and maximizing between class variance, an optimal discrimination boundary between classes in the training set is found.

Figure 4.7 is a plot of the 20 largest eigenvalues from a PCA decomposition in decreasing order. These values were generated from 222 facial range images (37 subjects under six facial expressions). As seen in figure 4.7, the twentieth eigenvalue is one thousand times smaller than the largest eigenvalue. In figure 4.8 we see the three eigenvectors (a.k.a., eigenfaces) associated with the three largest eigenvalues in descending order from left to right as images. These eigenfaces indicate the portions of the face where the most variability can be found. Lighter pixel values indicate regions of greater variation and darker pixel values indicate regions of less variation.

4.7 Identification

Identification begins by acquiring a new facial mesh (one not used in the generation of the eigenspace X described in section 4.6). The new mesh must be captured in the same environment in which the training data was captured; specifically, the position and orientation of the subject’s face with respect to the camera must be maintained. Once a single face mesh is captured it is projected into a range image, I (as described in section 4.3), aligned (as described in section 4.4), preprocessed (as described in section 4.5), and reduced to a low dimensional vector (equation 4.8.) In order to recognize people from their range images we have used the nearest neighbor criterion and the Euclidean metric on the reduced image space. A test image is projected into the subspace as shown in equation 4.8. We call this projection \hat{P} where $\hat{P} \in \Re^{10 \times 1}$. \hat{P} is then compared to the classified training images (P_1, P_2, \dots, P_k) using the Euclidean metric. d_i describes the distance of the target image from a training image P_i (equation 4.9.) The identity of the nearest training image (\hat{i}) is assigned to the test image (equation 4.10).

$$d_i = \|\hat{P} - P_i\|^2 \quad (4.9)$$

$$\hat{i} = \operatorname{argmin}_{i \in [1, 2, \dots, k]}(d_i) \quad (4.10)$$

4.8 Experiment and Results

To perform identification experiments we first need to create two sets of images: training and testing. Training images are used to generate an orthogonal basis (as described in section 4.6) into which each range image in the training data set is projected. This results in a low dimensional representation of each range image in the training data set. The orthogonal basis and low dimensional representations for each range image in the training data set are then stored for comparison later. Test images are a set of range images of faces we wish to identify. Any subject we wish to identify must have at least one facial range image in the training data set. The faces in the test range images need not have the same facial expressions as those in the training data set. Each test image is then reshaped as a column

Table 4.1. Each row in this table indicates the results of an experiment in identification using range images and PCA. The first column shows the number of training images used; the second column shows how many images were used in the test data set; the third column lists the percentage of correctly identified faces. In all experiments there is no intersection between the training and test data sets.

Training Images	Test Images	% Correctly Identified
185	37	94
148	74	94
111	111	92
74	148	84
37	185	74

Table 4.2. In these results 30 eigenvectors from PCA were used for projection with range images of size 242×347 (i.e., where 10 eigenvectors have previously been used for projection, we now use 30, and where range images were previously 41×57 pixels in size, they are now 242×347 .)

Training Images	Test Images	% Correctly Identified
185	37	94
148	74	93
111	111	92
74	148	88
37	185	82

vector (equation 4.6) and projected into the orthogonal basis (equation 4.8) resulting in a ten dimensional representation of the test image. Using the nearest neighbor algorithm from section 4.7, compare the test image’s ten dimensional representation to all other training image ten dimensional representations to find the identity of the subject in the test image.

The results shown in tables 4.1 through 4.5 were obtained using a selection of 222 and 492 meshes from 37 and 82 subjects respectively from the Laboratory for Computational Vision (LCV) facial mesh database. The column labeled "training" in each table shows the number of training images used to find the orthogonal basis; the column labeled "test" indicates how many faces were used in the matching test; successive columns give the percentage of correctly identified persons. If a test used 185 training images and 37 test images, then

Table 4.3. In these results 60 eigenvectors from PCA were used for projection with range images of size 242×347 .

Training Images	Test Images	% Correctly Identified
185	37	94
148	74	93
111	111	92
74	148	89
37	185	83

Table 4.4. For this experiment, 82 subjects with a total of 492 faces are used. Images are 41×61 pixels. All results are percentages. PCA (Principal Component Analysis), ICA (Independent Component Analysis) and FDA (Fisher’s Discriminate Analysis) are compared. Projections are 10, 30, or 60 dimensional as indicated. Algorithmic limitations lead to the missing results.

		10 Dimensions			30 Dimensions			60 Dimensions		
Training	Test	PCA	ICA	FDA	PCA	ICA	FDA	PCA	ICA	FDA
410	82	67.56	45.94	79.26	72.97	86.48	90.24	75.67	83.78	96.34
328	164	74.32	78.37	79.26	79.72	91.89	93.29	81.08	91.89	94.51
246	246	64.86	72.97	76.01	73.87	87.38	89.43	75.67	90.00	93.08
164	328	58.78	61.48	75.91	62.16	78.37	88.71	63.51	78.37	85.67
82	410	52.97	61.08	N/A	57.29	70.81	N/A	58.37	69.72	N/A

Table 4.5. For this experiment, 82 subjects with a total of 492 faces are used. Images are 202×306 pixels. All results are percentages. PCA (Principal Component Analysis), ICA (Independent Component Analysis) and FDA (Fisher’s Discriminate Analysis) are compared. Projections are 10, 30, or 60 dimensional as indicated. Computational and algorithmic limitations lead to missing results.

		10 Dimensions			30 Dimensions			60 Dimensions		
Training	Test	PCA	ICA	FDA	PCA	ICA	FDA	PCA	ICA	FDA
410	82	59.45	68.29	70.73	70.27		86.58	72.97		92.68
328	164	71.62	74.39	75.00	75.67	87.19	92.07	79.72		94.51
246	246	63.96	72.76	75.60	73.87	83.73	88.61	75.67	88.21	91.86
164	328	56.75	65.85	66.46	63.51	78.35	86.58	63.51	82.01	82.62
82	410	52.43		N/A	58.37		N/A	58.91		N/A

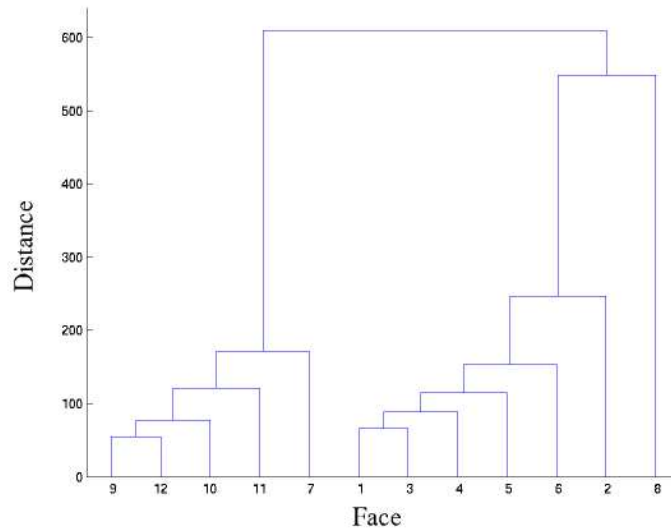


Figure 4.9. This dendrogram illustrates the Euclidean distance between the projections of twelve faces in the data set. Faces one through six belong to one person; faces seven through twelve belong to a different person. Using this method, we see that face eight is relatively far from both groups.

five facial expressions from each person were used in the training data set, and one facial expression from each person was used in the test data set. The set of training images and testing images have no intersection. In tables 4.2 and 4.3 we report the identification results achieved when the size of range images and the number of eigenvectors used for projection are increased. As these results indicate, increasing the size of range images and the number of eigenvectors used for projection augments identification by approximately 5%. Experimental results (not shown here) indicate that the increase in recognition rates are due more to the use of additional eigenvectors for projection than to the use of larger range images (larger than 41×57). As indicated in tables 4.4 and 4.5, FDA consistently gives the best recognition performance on this database. Additionally, we see again that increasing the size of range images does not aid the recognition performance.

The similarities between range images are illustrated in a dendrogram in figure 4.9. It illustrates the Euclidean distance between range images of twelve faces from two different

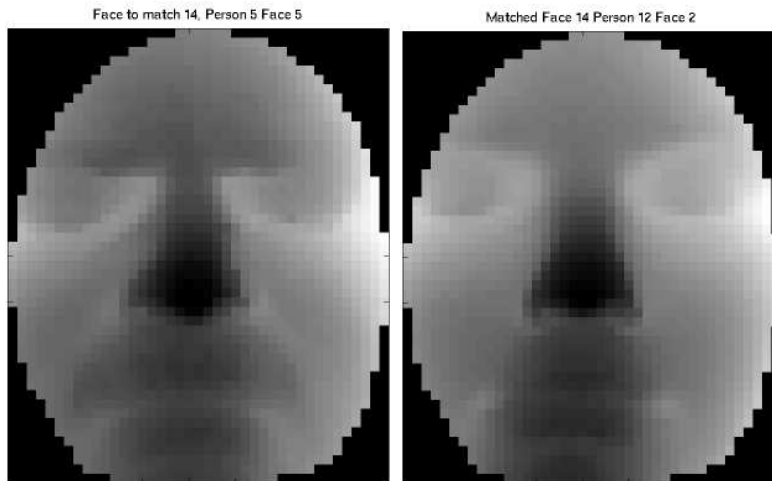


Figure 4.10. On the left is the range image of the person we wish to identify (person 5, expression 5). On the right is the incorrect result of the nearest-neighbor matching algorithm (person 12, face 2). This mismatch occurred when using three training faces per person.

persons after projection into the PCA eigenspace. Faces are numbered along the horizontal axis and distances are indicated on the vertical axis. Faces 1-6 are from one person and faces 7-12 are from a second person. A potential recognition problem can be seen with face 8 since its projection is relatively far from all other projections. Not all matches are successful. An example of two faces incorrectly matched can be seen in figure 4.10. In the left of figure 4.10 we see the range image for the face we wish to identify, and on the right we see the face (incorrectly) identified as coming from the same person.

4.9 Conclusions and Future Work

The aforementioned results show that range images are an effective means of identification. As shown in tables 4.1 through table 4.5 the percentage of correctly identified persons is better than what random selection would give, even for the smallest training data set (37 faces).

The database used for these experiments holds 690 facial meshes for 115 unique persons. Due to difficulties with registration, data capture, and the varying size of individual faces,

two subsets of 37 subjects (222 facial meshes) and 82 subjects (492 facial meshes) were used. It is not known to what extent the facial meshes captured in this database represent the range of facial deformations induced by facial expression. Limitations in computational resources also limit the size of range images that can be effectively used therefore reducing the accuracy of range images. Also, our current use of non-robust reduction techniques does not work well with noise in the data (possibly induced by error in the mesh capture or reduction techniques.) Finally, for certain individuals, the variability induced by facial expressions in facial range imaging is greater than the variability induced by identity leading to the outlier viewable as face eight in figure 4.9.

Among PCA, ICA, and FDA, FDA is the best reduction technique for this data set. With additional research into non-linear reduction techniques, better extra-class distinction could be made. A new method of space reduction such as non-linear Independent Component Analysis (ICA) or Fisher discriminates should be researched. Finding a better reduction method should improve recognition. A larger data set should also be used to test the usefulness of range images in a real world setting. Although the current results do not support it, increasing the size of range images used in training and testing could allow for the discovery of more subtle differences between individuals. Range imaging alone is not likely to solve the identification problem. The use of other imaging modalities (infrared, reflectance) combined with range imaging is more likely to lead to a robust system for identification.

CHAPTER 5

TOOLS FOR DERIVING A STATISTICAL MODEL OF RANGE IMAGES

Due to recent advances in imaging technology, range imaging (as well as infrared imaging) is now a practical alternative to more traditional visible spectrum imaging methods. Also of note, range imaging does not inherently depend on lighting or textures (two large sources of variation in visible spectrum imaging modalities.) Range images capture the distance and partial shapes of imaged objects irrespective of lighting or texture. Because of this, individuals now have the opportunity to build 3D models from range image observations since the variability in the image is a direct result of the shapes of imaged objects[22]. If the shapes of objects in the scene are anticipated, as is the case in automated target recognition, that knowledge can guide investigations into the image. While some work has been done in the image analysis field on range images, a formal statistical framework for range image understanding remains to be investigated [22]. Grenander and Srivastava have proposed that objects in a scene may be categorized by observing several discrete distance measurements within a scene [22]. As an example, one such measurement can be seen in the left image of figure 5.1. The distance between the range camera and the imaged object, along a particular direction, provides a range measurement. By placing a probability on the values associated with certain view vectors, objects in the scene can be recognized (i.e., perhaps we can state that the object in the scene is a plane, but we cannot say what type of plane).

As described in [22], the generation of probability models requires computation of a map, termed intersection map here, for each object and each imaging direction. Our goal in this thesis is to develop a software that generates the intersection maps. In general, intersection maps are generated by observing the intersection of a mesh with a vector projecting from

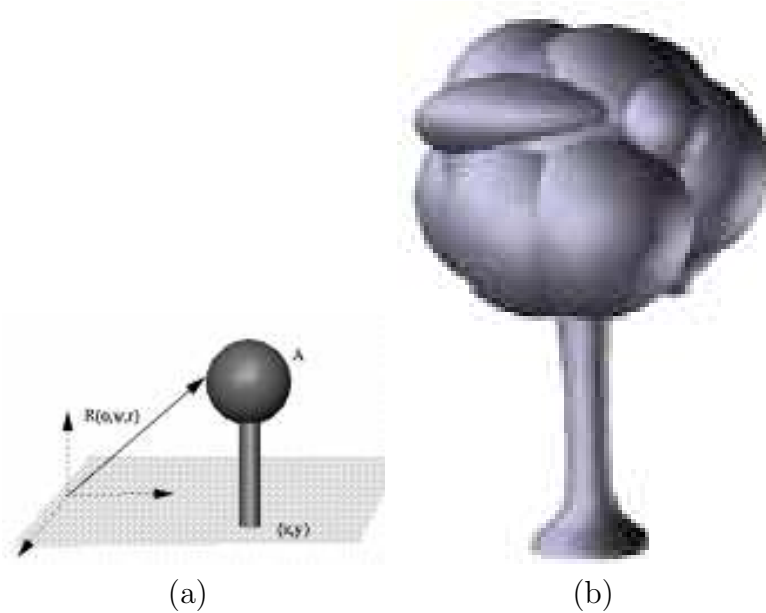


Figure 5.1. (a) An illustration of the vector from a view point to a location in the scene. Intersections of the tree object in the scene with the vector are of interest. (b) The tree mesh used in the generation of the intersection maps shown in figure 5.2.

a view point to a location in the viewed scene. As the mesh of an object translates in a plane, certain positions of that object lead to an intersection of the vector and the object. This position is marked as a white pixel in the map where the map pixels correspond to all possible translations of the object in the scene. In figure 5.1 (a) an illustration of the vector from a view point to a location in the scene is shown. Intersections of the tree object in the scene with the vector are of interest. Possible maps generated by this procedure using the tree mesh shown in figure 5.1 (b) are shown in figure 5.2. For applications of these intersection maps in the derivation of probability models and their applications in object classification, please refer to [22].

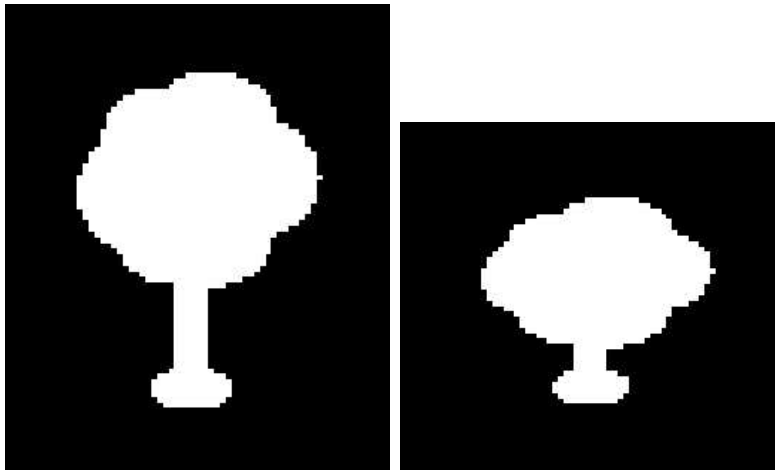


Figure 5.2. Two intersection maps.

APPENDIX A

DERIVATION OF $\frac{\partial}{\partial q_i} C(\mathbf{Q})$

Let $\mathbf{x} = x_1, x_2, x_3$, $\boldsymbol{\theta} = \theta_1, \theta_2, \theta_3$ and $\mathbf{Q} = (\mathbf{x}, \boldsymbol{\theta}) = \{q_i\}$

$$C(\mathbf{Q}) = \sum_m \sum_n (\mathbf{T}_k - \mathbf{E}(\mathbf{Q}))^2 \quad (\text{A.1})$$

$$\frac{\partial}{\partial q_i} C(\mathbf{Q}) = \frac{\partial}{\partial q_i} \left(\sum_m \sum_n (\mathbf{T}_k - \mathbf{E}(\mathbf{Q}))^2 \right) \quad (\text{A.2})$$

$$\frac{\partial}{\partial q_i} C(\mathbf{Q}) = \sum_m \sum_n \left((-2) (\mathbf{T}_k - \mathbf{E}(\mathbf{Q})) \left(\frac{\partial \mathbf{E}(\mathbf{Q})}{\partial q_i} \right) \right) \quad (\text{A.3})$$

$$\frac{\partial}{\partial q_i} C(\mathbf{Q}) = -2 \left\langle \mathbf{T}_k - \mathbf{E}(\mathbf{Q}), \frac{\partial \mathbf{E}(\mathbf{Q})}{\partial q_i} \right\rangle \quad (\text{A.4})$$

Where $\langle A, B \rangle$ indicates the cross product between two matrices A and B .

REFERENCES

- [1] Arvo, J. *Graphics Gems II*. Academic Press, Inc., 1991.
- [2] Balakrishnama, S., A. Ganapathiraju. *Linear Discriminant Analysis - A Brief Tutorial*. Institute for Signal and Information Processing, Dept. of Electrical and Computer Engineering, Mississippi State Univeristy.
- [3] Barrett, William A. *A Survey of Face Recognition Algorithms and Testing Results*. Conference Record of the Thirty-First Asilomar Conference on Signals, Systems, and Computers, pp. 301-305, 1998.
- [4] Belhumeur, P. N., J. P. Hespanha, D. J. Kriegman. *Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19(7), pp. 711-720, 1997.
- [5] Bregler, Christoph, Jitendra Malik. *Tracking People with Twists and Exponential Maps*. Proc. Computer Vision and Pattern Recognition, pp. 8-15, 1998.
- [6] Brigham, J. C., A. Maass, L. D. Snyder, K. Spaulding. *Accuracy of Eyewitness Identification in a Field Setting*. Journal of Personality and Social Psychology, vol. 42, pp. 673-681, 1982.
- [7] Brown, E., K. Deffenbacher, W. Sturgill. *Memory for Faces and the Circumstances of Encounter*. Journal of Applied Psychology, vol. 62, pp. 311-318, 1977.
- [8] Brunelli, Roberto, Thomaso Poggio. *HyperBF Networks for Gender Classification*. Proc. DARPA Image Understanding Workshop, pp. 311-314, 1992.
- [9] Brunelli, Roberto, Daniele Falavigna. *Person Identification using Multiple Cues*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 10, October 1995.
- [10] Castleman, Kenneth R. *Digital Image Processing*. Prentice Hall, Inc., 1996.
- [11] Chellappa, Rama, Charles L. Wilson, Saad Sirohey. *Human and Machine Recognition of Faces: A Survey*. Proc. IEEE, vol. 83, no. 5, pp. 705 - 740, May 1995.
- [12] Daugman, John. *Face and Gesture Recognition: Overview*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, issue 7, July 1997.
- [13] Foley, James D., Andries van Dam, Steven K. Feiner, John F. Hughes. *Computer Graphics Principles and Practice, Second Edition*. Addison-Wesley Publishing Company, Inc., 1990.

- [14] Fromherz, Thomas. *Face Recognition: A Summary of 1995-1997*. Technical Report ICSI TR-98-027, International Computer Science Institute, Berkeley, 1998.
- [15] Galton, Francis. *Personal Identification and Description*. Nature, pp. 173-177 June 21, 1888, Nature, pp. 201-202, June 28, 1888.
- [16] Gavrilu, D. M., L. S. Davis. *3D Model-Based Tracking of Humans in Action: A Multi-View Approach*. Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 73-80, San Francisco, U.S.A., 1996.
- [17] Golomb, Beatrice Alexandra, Terrence J. Sejnowski. *SEXNET: A Neural Network Identifies Sex from Human Faces*. Advances in Neural Information Processing Systems 3, D. Touretzky, R. Lipmann, San Mateo, CA: Morgan Kaufmann, pp. 572-577, 1991.
- [18] Gong, Shaogang, Stephen J. McKenna, Alexandra Psarrou. *Dynamic Vision: From Images to Face Recognition*. Imperial College Press, 2000.
- [19] Gonzalez, Rafeal C., Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1992.
- [20] Gordon, Gaile G. *Face Recognition Based on Depth and Curvature Features*. Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, Illinois, pp. 108-110, June 1992.
- [21] Gurney, Kevin. *An Introduction to Neural Networks*. UCL Press Ltd., 1997.
- [22] Grenander, Ulf, Anuj Srivastava, Curt Heshner. *Models for Statistical Analysis of Range Images*. In review for Advances in Applied Statistics, November 2001.
- [23] Guo, Guo-Dong, Hong-Jiang Zhang. *Boosting for Fast Face Recognition*. Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems (RATFG-RTS 2001), Vancouver, Canada, July 13, 2001.
- [24] Guo, Guodong, Stan Z. Li, Kapluk Chan. *Face Recognition by Support Vector Machines*. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, March 28-30, 2000.
- [25] Guo, Guo-Dong, Hong-Jiang Zhang, Stan Z. Li. *Pairwise Face Recognition*. Eighth IEEE International Conference on Computer Vision, ICCV2001, vol. II, pp. 282-287, 2001.
- [26] Gutta, S., J. Huang, B. Takacs, H. Wechsler. *Face Recognition Using Ensembles of Networks*. International Conference on Pattern Recognition (ICPR), Vienna, Austria, 1996.
- [27] Hallinan, Peter L., Gaile G. Gordon, Alan L. Yuille, Peter Giblin, David Mumford. *Two and Three-Dimensional Patterns of the Face*. A. K. Peters Ltd., May 1999.
- [28] Haritaoglu, Ismail, David Harwood, Larry S. Davis. *W⁴: Who? When? Where? What? A Real Time System for Detecting and Tracking People*. International Conference on Face and Gesture Recognition, April 14-16, Nara, Japan, 1998.

- [29] Harmon, L., W. Hunt. *Automatic Recognition of Human Face Profiles*. Computer Graphics and Image Processing, vol. 6, pp. 135-156, 1977.
- [30] Hochberg, Julian, Ruth Ellen Galper. *Recognition of Faces: I. An Exploratory Study*. Psychonomic Science, vol. 9, pp. 619-620, 1967.
- [31] Hotelling, H. *Analysis of a Complex of Statistical Variables into Principal Components*. J. Educ. Psychol., 24, pp. 417-441, pp. 498-520, 1933.
- [32] Howell, A. Jonathan, Hilary Buxton. *Towards Unconstrained Face Recognition from Image Sequences*. Proceedings of the International Conference on Automatic Face and Gesture Recognition, Killington, VT, pp. 224-229, 1996.
- [33] Hyvärinen, Aapo. *Survey on Independent Component Analysis*. Neural Computing Surveys, vol. 2, pp. 94-128, 1999.
- [34] Hyvärinen, Karhunen, Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
- [35] Hyvärinen, Aapo, Erkki Oja. *Independent Component Analysis: A Tutorial*. http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/, June 2, 2003.
- [36] Jolliffe, I. T. *Principal Component Analysis*. Springer-Verlag New York Inc., 1986.
- [37] Jonsson, K., J. Matas, J. Kittler, Y. P. Li. *Learning Support Vectors for Face Verification and Recognition*. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, March 28-30, 2000.
- [38] Kanade, T. *Computer Recognition of Human Faces*. Basel and Stuttgart: Birkhauser, 1977.
- [39] Karhunen, K. *Über Lineare Methoden in der Wahrscheinlich-Keitsrechnung*. Ann. Acad. Sci. Fennicae, Ser. A.I.37 (English translation by I. Selin, *On Linear Methods in Probability Theory*. Doc. T-131, The RAND Corp., Santa Monica, CA, 1960), 1947.
- [40] Kaufman Jr., G., K. Breeding. *The Automatic Recognition of Human Faces from Profile Silhouettes*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 6, issue 2, pp. 113-121, 1976.
- [41] Kaya, Y., K. Kobayashi. *A Basic Study on Human Face Recognition*. Frontiers of Pattern Recognition, S. Watanabe, Ed. New York: Academic, 1972.
- [42] Kelly, M. *Visual Identification of People by Computer*. Technical Report Ai-130, Stanford AI Project, Stanford, CA, 1970.
- [43] La Cascia, Marco, Stan Sclaroff, Vassilis Athitsos. *Fast, Reliable, Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models*. Boston University, Computer Science Technical Report 99-005, May 1999.

- [44] Lades, Martin, Jan C. Vorbrüggen, Joachim Buhmann, Jörg Lange, Christoph von der Malsburg, Rolf P. Würtz, Wolfgang Konen. *Distortion Invariant Object Recognition in the Dynamic Link Architecture*. Transactions on Computers, vol. 42, no. 3, March 1993.
- [45] Li, Yongmin, Shaogang Gong, Heather Liddell. *Video-Based Face Recognition Using Identity Surfaces*. Proc. Second International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Realtime Systems, July 2001.
- [46] Lipton, Alan J., Hironobu Fujiyoshi, Raju S. Patil. *Moving Target Classification and Tracking from Real-time Video*. IEEE Workshop on Applications of Computer Vision (WACV), pp. 8-14, Princeton NJ, October 1998.
- [47] Liu, Xiuwen, Anuj Srivastava, Kyle Gallivan. *Optimal Linear Representations of Images for Object Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, under review, 2003.
- [48] Loève, M. *Fonctions Aléatoires de Second Ordre*. in P. Lévy, *Processus Stochastiques et Mouvement Brownien*. Hermann, Paris, 1948.
- [49] Lu, Juwei, K. N. Plataniotis, A. N. Venetsanopoulos. *A Kernel Machine Based Approach for Multi-View Face Recognition*. IEEE 2002 International Conference on Image Processing, September 22-25, 2002.
- [50] Marcialis, Gian Luca, Fabio Roli. *Fusion of LDA and PCA for Face Recognition*. <http://www-dii.ing.unisi.it/aiia2002/paper/PERCEVISIO/Abstract/marcialis1-aiia02.pdf>, January 22, 2003.
- [51] Maurer, Thomas, Christoph von der Malsburg. *Single-View Based Recognition of Faces Rotated in Depth*. Proc. International Workshop on Face and Gesture Recognition, pp. 176-181, 1996.
- [52] Non-contact 3D Digitizer Vivid 700/VI-700, Instruction Manual (Hardware).
- [53] O'Toole, A. J., H. Abdi, K. A. Deffenbacher, D. Valentin. *Low-dimensional Representation of Faces in Higher Dimensions of the Face Space*. Journal of the Optical Society of America A, vol. 10, no. 3, pp. 405-410, March 1993.
- [54] Phillips, P. J., H. Wechsler, J. Huang, P. Rauss. *The FERET Database and Evaluation procedure for Face Recognition Algorithms*. Images and Vision Computing, vol. 16, pp. 295-306, 1998.
- [55] Poggio, Thomaso, Federico Girosi. *A Theory of Networks for Approximation and Learning*. July 1989.
- [56] *Proceedings of the Eighth IEEE International Conference on Computer Vision*. July 7-14, 2001.
- [57] Rizvi, S., P. J. Phillips, H. Moon. *The FERET Verification Testing Protocol for Face Recognition Algorithms*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, issue 10, pp. 1090-1104, October 2000.

- [58] Rizvi, S. A., P. J. Phillips, H. Moon. *A Verification Protocol and Statistical Performance Analysis for Face Recognition Algorithms*. Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 833-838, 1998.
- [59] Rosenberg, Yoav, Michael Werman. *Real-Time Object Tracking from a Moving Video Camera: A Software Approach on a PC*. IEEE Workshop on Applications of Computer Vision, pp. 238-239, Princeton, October 1998.
- [60] Samal, A., P. A. Iyengar. *Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey*. Pattern Recognition, vol. 25, no. 1, pp. 65-77, 1992.
- [61] Schödl, Arno, Antonio Haro, Irfan A. Essa. *Head Tracking Using a Textured Polygonal Model*. Workshop on Perceptual User Interfaces, pp. 43-48, San Francisco, California, November 1998.
- [62] Sirovich, L., M. Kirby. *Low-dimensional Procedure for the Characterization of Human Faces*. Journal of the Optical Society of America A, vol. 3, pp. 519-524, 1987.
- [63] Soucek, B. John G. Sutherland, G. Visaggio. *Holographic Decision Support System: Credit Scoring Based on Quality Metrics*. Plantamura, V.L. et al. ed., Frontier Decision Support Concepts, John Wiley & Sons, New York, 1994.
- [64] Srivastava, Anuj. *Bayesian Filtering for Tracking Pose and Location of Rigid Targets*. Proc. SPIE Aerosense, April 2000.
- [65] Srivastava, Anuj. *Computational Methods in Statistics*. <http://calais.stat.fsu.edu/~{anuj}/PDF-files/STA5106-f-02/book0.pdf>, January 22, 2003.
- [66] Srivastava, Anuj, Xiuwen Liu, Brian Thomasson, Curt Heshner. *Spectral Probability Models for IR Images and their Applications to Face Recognition*. Proc. of Computer Vision and Pattern Recognition workshop on Beyond Visual Spectrum, Hawaii, 2001.
- [67] Stonham, T. J. *Practical Face Recognition and Verification with WISARD*. Aspect of Face Processing, H. Ellis, M. Jeeves, F. Newcombe, A. Young, Dordrecht: Nijhoff, pp. 426-441, 1984.
- [68] Sutherland, John. G. *Holographic Model of Memory, Learning, and Expression*. International Journal of Neural Systems, vol. 1-3, pp. 256-257, 1990.
- [69] Sutherland, John G. *A Transputer Based Implementation of Holographic Neural Technology*. Transputing 91-2, 1991.
- [70] Sutherland, John G. *The Holographic Neural Method*. Soucek B. Ed. Fuzzy, Holographic and Parallel Intelligence: The Sixth Generation Breakthrough, John Wiley & Sons, New York, 1992.
- [71] Sutherland, John G. *The Holographic Cell. A Quantum Perspective*. Plantamura V.L. et al. ed. Frontier Decision Support Concepts, John Wiley & Sons, New York, 1994.

- [72] Turk, Matthew, Alex Pentland. *Face Recognition using Eigenfaces*. Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 586-591, Maui, Hawaii, 1991.
- [73] Valentin, Dominique, Hervé Abdi, Alice J. O’Toole, Garrison W. Cottrell. *Connectionist Models of Face Processing: A Survey*. Pattern Recognition, Vol. 27, no. 9, pp. 1209-1230, 1994.
- [74] Valentine, Tim. *Cognitive and Computational Aspects of Face Recognition: Explorations in Face Space*. Routledge, 1995.
- [75] Wiskott, Laurenz, Jean-Marc Fellous, Norbert Krüger, Christoph von der Malsburg. *Face Recognition by Elastic Bunch Graph Matching*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, July 1997.
- [76] Yang, Jie, Hua Yu, William Kunz. *An Efficient LDA Algorithm for Face Recognition*. Proceedings of the Third IEEE Workshop on Applications of Computer Vision (WACV-96), pp. 142-147, Sarasota, Florida, USA, December 1996.
- [77] Zhao, W. Y., Rama Chellappa, A. Rosenfeld, P. J. Phillips. *Face Recognition: A Literature Survey*. UMD CfAR Technical Report, CAR-TR-948, 2000.
- [78] AcSys Biometrics
<http://www.acsysbiometrics.com>
 January 15, 2003.
- [79] Alexei Aganin, Methods for 3D Registration and Recognition
<http://www.math.fsu.edu/~aaganin/>
 January 29, 2003.
- [80] American Civil Liberties Union
<http://www.aclu.org>
 November 5, 2002.
- [81] Berninger Software BmbH
<http://www.berningersoftware.de>
 January 15, 2003.
- [82] Biometrica Systems, Inc.
<http://www.biometrica.com>
 January 15, 2003.
- [83] Cognitec Systems
<http://cognitec-systems.de>
 January 15, 2003.
- [84] Dialog Communication Systems Inc.
<http://www.bioid.com.tw/>
 January 23, 2003.

- [85] Evaluation of Face Recognition Algorithms
<http://www.cs.colostate.edu/evalfacerec/index.html>
November 5, 2002.
- [86] Eyematic Interfaces, Inc.
<http://www.eyematic.com>
January 15, 2003.
- [87] Face Detection Home Page
<http://home.t-online.de/home/Robert.Frischholz/face.htm>
November 5, 2002.
- [88] The Face Recognition Home Page
<http://www.cs.rug.nl/users/peterkr/FACE/face.html>
November 5, 2002.
- [89] FaceKey Corp.
<http://www.facekey.com>
January 15, 2003.
- [90] Facial Recognition Projects, Department of Defense Counterdrug Technology Development Program Office
<http://www.dodcounterdrug.com/facialrecognition/>
January 23, 2003.
- [91] Humanscan GmbH
<http://www.bioid.com>
January 15, 2003.
- [92] Identix, Inc.
<http://www.identix.com>
January 15, 2003.
- [93] Imagis Technologies, Inc.
<http://www.imagistechnologies.com>
January 15, 2003.
- [94] International Biometric Group
<http://www.facial-scan.com>
November 5, 2002.
- [95] International Biometric Group
<http://www.biometricgroup.com>
November 5, 2002.
- [96] Keyware Technologies N.V.
<http://www.keyware.com>
January 15, 2003.

- [97] Neurodynamics Limited
<http://www.neurodynamics.com>
January 15, 2003.
- [98] RaindropGeomagic Decimate
<http://www.raindropgeomagic.com/products/decimate/>
January 23, 2003.
- [99] Viisage Technology
<http://www.viisage.com/facetools.htm>
January 22, 2003.
- [100] VisionSphere Technologies
<http://www.visionspheretech.com>
January 15, 2003.
- [101] ZN Vision Technologies
<http://www.zn-ag.com>
January 15, 2003.

BIOGRAPHICAL SKETCH

Matthew Curtis Heshel

Matthew (Curt) Heshel was born in Colorado Springs, Colorado on November 9, 1976. His family eventually settled in Port Orange, Florida where he attended grade school at Spruce Creek Elementary, Silver Sands Middle, and Spruce Creek High. He was admitted to the International Baccalaureate (IB) program during high school and was accepted at Furman University to work on his undergraduate education. After two years at Furman he transferred to Florida State University where he finished his masters degree in Computer Science. His current interests include identification and tracking of people in unconstrained environments.