OBJECT RECOGNITION IN SUBSPACES: APPLICATIONS IN BIOMETRY AND 3D MODEL RETRIEVAL

by

Helin Dutağacı

BS, in Electrical and Electronic Engineering, Boğaziçi University, 1999

MS, in Electrical and Electronic Engineering, Boğaziçi University, 2002

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in
Boğaziçi University
2009

OBJECT RECOGNITION IN SUBSPACES: APPLICATIONS IN BIOMETRY AND 3D MODEL RETRIEVAL

APPROVED BY:

Prof. Bülent Sankur (Thesis Supervisor)	.....
Assoc. Prof. Burak Acar	.....
Prof. Lale Akarun	.....
Assoc. Prof. Cem Ünsalan	.....
Assoc. Prof. Yücel Yemez	.....

DATE OF APPROVAL: 20.01.2009

To my parents, Hanife and Hıdır Dutağacı ...

ACKNOWLEDGEMENTS

I am grateful to Bülent Sankur for his infinite support, encouragement and patience. He devoted himself to the success of his PhD students with great enthusiasm, and I am fortunate to be one of them. I always admired his intelligence, vast scientific knowledge, and high ethical values.

I have special thanks to Yücel Yemez for joyfully attending our group meetings and bringing his humorous touch to our technical discussions. His tender understanding gave me strength along the way.

I would like to thank Lale Akarun for her affectionate support during my graduate study and scientific contributions to this thesis. I would like to thank Burak Acar and Cem Ünsalan for participating in my thesis committee and providing invaluable feedback. I am grateful to Cem Ünsalan for proofreading the thesis. I would also like to thank Afzal Godil for his support during the last months of my graduate study.

I would like to thank Francis Schmitt for his scientific guidance and for sharing his brilliant ideas during my visit to Paris and during his visit to Istanbul. The memory of Francis Schmitt will always be with the 3D research community.

Without the scientific contributions of my colleagues and friends Erdem Yörük, Berk Gökberk, and Ceyhun Burak Akgül, this thesis would have been impossible. Many thanks.

I would like to thank my eternal friend Koray Çiftçi for just existing in the lab next to my lab. That was all I needed to endure the difficulties of graduate school.

I would like to thank my professors Kadri Özçaldıran, Yorgo Istefanopulos, Günhan Dündar, Yağmur Denizhan, Hakan Deliç, Ayşın Ertüzün, Levent Arslan,

and Emin Anarım. It was a great honor to be their student.

I would like to thank Ferize Gözüm, who fascinated me with her endless scientific curiosity, for pushing me to finish this thesis.

During my graduate study, I have met with fabulous people who supported me both professionally and personally. They made the lab and the university a warm and fun place. Many thanks to İpek Şen, Ebru Arısoy, Doğaç Başaran, Hatice Çınar, Oya Çeliktutan, Luca Teijeiro Mosquera, Erinç Dikici, Erdem Yörük, Ceyhun Burak Akgül, Ender Konukoğlu, Sergül Aydöre, Yücel Altuğ, Sinan Yıldırım, Cem Demirkır, Çaglayan Dicle, Neslihan Gerek, Sıddıka Parlak, Nazlı Güney, Oya Aran, Berk Gökberk, Pınar Santemiz, Neşe Alyüz, Ali Albert Salah, Arman Savran, Özgür Devrim Orman, Barış Özgül, Ekin Şahin, and Elif Sürer Köse. I also thank Murat Saraçlar for his cheerful presence in the lab and his smart jokes. I regret I was too early to attend his courses.

I am grateful to my friends Özüm Seda Duran, Füsun Karaman, Sedef Özge, Evrim Dutağacı, Yılmaz Mete, Erkin Özalp, Zahit Atam, Çiçek Çavdar, Ayça Çiftçi, Elif Özsoy, Oya Benlioğlu, Ömer Gözüm, Meltem Başbuğ, Derya Koç, Xiaolan Li, and Gülay Dincel for bringing meaning, joy, and wisdom to my life during my long years of research.

This thesis is dedicated to my parents, Hanife and Hıdır Dutağacı. I have no words to express my gratitude to them. My sisters, Sevgi and Berçem Dutağacı, guided me whenever I was lost, supported me whenever I felt exhausted, made me smile whenever I was upset, and served me a cup of coffee whenever I needed it.

And my cat, Tırmık Mesih, the cutest and smartest creature in the world, thank you very much.

ABSTRACT

OBJECT RECOGNITION IN SUBSPACES: APPLICATIONS IN BIOMETRY AND 3D MODEL RETRIEVAL

Shape description is a crucial step in many computer vision applications. This thesis is an attempt to introduce various representations of two and three dimensional shape information. These representations are aimed to be in homogeneous parametric forms in 2D or 3D space, such that subspace-based feature extraction techniques are applicable on them. We tackle three different applications: (i) Person recognition with hand biometry, (ii) Person recognition with three-dimensional face biometry, (iii) Indexing and retrieval of generic three-dimensional models. For each application, we propose various combinations of shape representation schemes and subspace-based feature extraction methods. We consider subspaces with fixed bases such as cosines, complex exponentials and tailored subspaces such as Principal Component Analysis, Independent Component Analysis and Nonnegative Matrix Factorization.

Most of the descriptors we propose are dependent on the pose of the object. In this thesis we give special emphasis on the pose normalization of objects. This challenging step is highly application-specific. For hands and 3D faces, anatomical landmarks are used in order to reduce within-class variations due to pose, expression and articulation, whereas generic 3D models lack common landmarks. In order to deal with this disadvantage of generic models, we propose solutions that operate both in the pre-processing stage and in the matching stage.

ÖZET

ALTUZAYLARDA NESNE TANIMA: BİYOMETRİ VE 3B MODELLERİN GERİ GETİRİLMESİ UYGULAMALARI

Şekil tanıma, bilgisayarlı görü uygulamalarının önemli bir adımıdır. Bu tez, iki ve üç boyutlu şekil bilgisi için çeşitli gösterimler önermektedir. Bu gösterimlerin, altuzay tabanlı öznitelik çıkarımına uygun olması için 2B ve 3B uzayda birörnek parametrik formlarda olması amaçlanmıştır. Üç farklı uygulama üzerinde çalışılmıştır: (i) El biyometrisine dayalı kişi tanıma, (ii) Üç boyutlu yüz biyometrisine dayalı kişi tanıma, (iii) Üç boyutlu genelgeçer nesnelerin indekslenmesi ve geri-çağırımı. Her bir uygulama için, şekil gösterimlerinin ve altuzay tabanlı özniteliklerin çeşitli kombinasyonları denenmiştir. Kullanılan altuzay tabanlı yöntemler, kosinüsler ya da karmaşık üsseller gibi sabit tabanlarla betimlenen altuzayları içerebilir, asal bileşenler analizi, bağımsız bileşenler analizi ve negatif olmayan matris ayrıştırması gibi analizlere dayanabilir.

Önerilen betimleyicilerin çoğu nesnenin pozuna bağımlıdır. Bu tezde, nesnelerin poz düzgelenmesine özel bir önem verilmiştir. Poz düzgelemesi uygulamaya göre farklılık gösterir. El şekilleri ve 3B yüzlerde, poz, ifade ya da boğumlanma farklılarını gidermek için anatomik nirengi noktalarından faydalanılmıştır. Genelgeçer 3B nesneler ise ortak anatomik nirengi noktalarından yoksundur. Genel-geçer 3B nesnelerin bu dezavantajını gidermek için, gerek ön işleme gerekse karşılaştırma aşamalarında kullanılmak üzere çeşitli çözümler önerilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

$\mathbf{A}$	Matrix of mixing coefficients
$\mathbf{b}$	Coefficient vector
$\mathbf{C}$	Covariance matrix
$d(\cdot,\cdot)$	Distance metric
$\mathbf{D}$	Distance matrix
$DT_f(\cdot)$	Distance transform of $f(\cdot)$
$f(\cdot)$	3D binary function
$\mathbf{H}$	Nonnegative coefficient matrix in NMF
N	Dimension of the original space
p	3D point
R	Resolution of voxel representation
$\mathbf{S}$	Matrix of source signals
$\mathbf{u}_i$	Eigenvector
$\mathbf{U}$	Matrix of eigenvectors
$\mathbf{V}$	Nonnegative basis vectors of NMF space
w_N	N'th root of unity
$\mathbf{W}$	Separating matrix
$\mathbf{x}$	Data vector
$\mathbf{X}$	Data matrix
$\mathbf{z}$	Hand contour vector
α	Texture to shape ratio
θ	Angular coordinate in polar coordinate system
λ_i	Eigenvalue
μ	Mean data vector
ρ	Radial coordinate in polar coordinate system
$\mathbf{\Phi}$	Matrix of basis vectors
σ	Width of the Gaussian profile

AFM	Average face model

ARR	Axis relabeling and reflection
ART	Angular radial transform
AWMD	Area weighted mean distance
CAD	Computer aided design
CAM	Computer aided manufacturing
CbARR	Class based axis relabeling and reflection
CPCA	Continuous principal component analysis
CRSP	Concrete radialized spherical projection
DBF	Density based framework
DCG	Discounted cumulative gain
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DT	Distance transform
EER	Equal error rate
EGI	Extended Gaussian image
FAR	False acceptance rate
FRGC	Face recognition grand challenge
FRR	False rejection rate
FT	First tier
GDT	Gaussian of distance transform
ICA	Independent component analysis
ICP	Iterative closest point
IDT	Inverse of distance transform
KLT	Karhunen-Loève transform
LDA	Linear discriminant analysis
MbARR	Mean shape based axis relabeling and reflection
LDT	Linear function of distance transform
LFD	Light field descriptor
MDS	Multidimensional scaling
NMF	Nonnegative matrix factorization
NN	Nearest neighbor

NURBS	Non-uniform rational B-spline
PCA	Principal component analysis
PSB	Princeton shape benchmark
ROC	Receiver operating characteristics
SFBS	Sequential floating backward search
ST	Second tier
TPS	Thin-plate spline

1. INTRODUCTION

Shape-based object recognition is a broad research area encompassing many domains such as medical imaging, biometry, bioinformatics, archeology, astronomy, industrial inspection, quality control, robot vision, and so on. There is also great diversity of the data structures that represent the ”shape” of real objects through acquisition and reconstruction, virtual objects through modeling and a combination of both through post-processing and modification. Each application has its own notion of ”shape” and has specific demands from shape-based computer vision algorithms.

According to the definition of Kendall [1], the shape of a subset of a Euclidean space is all the geometric information that remains under similarity transformations (translation, scaling and rotation). In the domain of computer vision, the invariance to similarity transformations is the most emphasized requirement of a shape recognition system. The required invariance can even be extended to affine transformations where the geometry of an object is mapped onto two-dimensional images and distorted by perspective transformation during acquisition. In other cases, invariance to articulation (hand recognition) or certain deformations (expression-invariant face recognition) is of great importance since such variations do not alter the ”identity” of the object.

The ultimate desired property of shape recognition system is the discriminating power. This requirement is strongly related to the ”invariance property” since similarities between two different objects can be significantly higher than the similarities between different transformations (similarity, affine, articulation, deformation) of the same object.

The discriminative power of a shape descriptor depends on the definition of ”identity” or ”relevance” and the notion of similarity. For example, shape descriptors used for the purpose of 3D face recognition are desired to be invariant to facial

expressions and be sensitive to personal features. The smiling and crying face of the same person should be labeled with the same identity, while neutral faces of two different individuals should be considered totally different even if the two individuals are identical twins. If the application at hand is the retrieval of generic 3D models then two human faces of any individuals with any facial expression are considered as relevant and their mutual similarity score should be high. It is apparent that the relation between geometric information and identity and the notion of "shape similarity" is highly application dependent.

Other desired properties of shape-based recognition systems are invariance to resolution, operability under unorganized and noisy data, and efficiency in terms of the complexity and speed of the algorithm and the storage size of the description.

Kendall's [1] notion of shape (or preshape) is in general applicable to any dimension, although the practical shape is two or three-dimensional. The research on 2D shape analysis is tremendous and goes back as far as the emergence of computer vision [2, 3]. Research on 3D shape analysis is relatively new. The earliest recognition algorithms deal primarily with range data [4]. In the last two decades, studies for shape-based description and matching of complete free-form 3D models have progressed in different fields such as medical imaging, recognition of CAD/CAM models, indexing and retrieval of generic objects.

The data structure containing the geometric information of an object is of great importance. There are a number of ways to represent 2D shapes (silhouettes, contours, sets of 2D points, polygon approximations, splines) and even more diverse data structures for 3D shapes (point clouds, polygonal meshes, NURBS, solid models, voxel structures, range images, etc.). Some of these structures are parametric functions, such as silhouettes, regularly sampled contours, voxelized models and depth maps. They can be processed by conventional signal processing tools. Others are difficult to be interpreted as parametric signals such as point sets and polygonal meshes. Converting such structures into a parametric form is beneficial since there will be a common domain and coordinates for shape representation. This form en-

ables us to cast the geometric information of the object of interest into a fixed-length vector with well-defined coordinates.

As soon as the shape is converted into a vector of fixed-length N, it can be regarded as a point lying in the N-dimensional Euclidean space. Since the elements of this vector are highly correlated, we can safely assume that the volume occupied by the objects of interest in this Euclidean space has a certain structure. That is the main motivation behind the use of subspace analysis as a tool to characterize shape and to recognize objects.

In this thesis, we concentrate on three different applications of shape recognition: (i) Hand recognition, (ii) 3D face recognition, and (iii) Retrieval of generic 3D models. Inputs to these applications are 2D shape, range image (2 1/2-D shape) and complete 3D shape model, respectively. For each of these inputs, we consider a number of techniques to represent the geometric information in vector forms that are suitable for subspace-based analysis. Then we compare the recognition and retrieval performances of various subspace techniques operating on these representations. Some combinations of the representation types and subspace-based schemes were previously applied to these problems, and some are first considered in this study. Motivations and challenges regarding each of these three applications, and our contributions to advance the state-of-the-art of these applications will be described in detail in the proceeding chapters.

The thesis is organized as follows: In Chapter 2, we review the subspace-methods we have used in this study. We make distinctions between model-based subspaces and data-driven subspaces as well as reconstructive and discriminative subspaces.

In Chapter 3, we provide an extensive survey on the state-of-the-art of hand-based biometry, we bring new approaches to the field and give a detailed experimental evaluation on our large hand database. We compare several subspace-based feature sets on the normalized hand shape and appearance. We emphasize the

importance of hand normalization in order to make the subspace-based shape modeling free of intra-person variations of global positioning and finger articulation. We explore many parameters of the hand-based biometry, such as use of left and right hands or of ambidextrous access, the choice of acquisition devices, the impact of time lapse, resolution and the size of the training set.

In Chapter 4, we investigate recognition performances of various subspace-based features applied on registered 3D face scans. We apply the feature extraction techniques to different representations of registered faces, such as 3D point clouds, 2D depth images and 3D voxel. We consider both global and local features. Global features are extracted from the whole face data, whereas local features are computed over the blocks partitioned from 2D depth images. Experiments using different combinations of representation types and feature vectors are conducted on the 3D-RMA dataset and the FRGC face database.

In Chapter 5, we propose the application of masks as a means to mitigate expression-distortions on 3D faces and to enhance their recognition performance. Masking becomes necessary to de-emphasize the face regions that deform under expression. We first show that warping the depth values of corresponding face points onto the same spatial coordinates while obtaining the 2D depth images is beneficial, and second, that proper masking can improve the recognition performance.

In Chapter 6, we study the indexing and retrieval of generic 3D models. We present a retrieval scheme based comparatively on three subspaces, PCA, ICA and NMF, extracted from the volumetric representations of 3D models. We find that the most propitious 3D distance transform leading to discriminative subspace features is the inverse distance transform. We mitigate the ambiguity of pose normalization with continuous PCA, by the use of all feasible axis labeling and reflections. The performance of the subspace-based retrieval methods on Princeton Shape Benchmark is on a par with the state-of-the-art methods.

Each chapter has its own concluding section regarding the achievements of this

work and future perspectives for the specific application in hand. However, we give a summary of our results in Chapter 7.

2. SUBSPACES

Subspace methods have been widely used for dimensionality reduction and feature extraction. They are popular to analyze structures among data in diverse domains such as engineering, economics, astronomy, biology, psychology, almost in every field where large amount of correlated numerical data are available.

In the context of shape-based object recognition, we ask the following questions:

- How should we represent the shape information?
- Which subspace methods should we use?
- Should we modify the shape representations to be suitable for subspace analysis? This question addresses tasks such as pose-normalization, alignment of shapes and matching strategies.

While attempting to answer these questions, we focus on three specific applications: (i) Hand shape-based biometry, (ii) 3D face-based biometry, (iii) indexing and retrieval of 3D generic objects.

The choice or design of $\mathbf{x}$, which is the vector representing the geometric information, is extremely important since it determines the space of the patterns we are dealing with and the distribution of the patterns on this space. For example, gray-valued images of size $N_1 \times N_2$ can be represented as vectors (or points) that lie in the N-dimensional Euclidean space, $\mathbb{R}^N$, where $N = N_1 \times N_2$. Each pixel location corresponds to an axis of this space and the intensity of a particular image at that pixel location is the coordinate of the N-dimensional point representing the image. The N_1 sample points of contours of 2D binary shapes lie in the $2 \times N_1$-dimensional Euclidean space. In the case of hand recognition, contour-based representations and silhouette-based representations lie in totally different spaces and they exhibit very different behaviour in these spaces. Most of the time, it is the original representation, rather than the choice of a particular subspace technique that determines the

performance of a recognition system.

We leave the first question regarding the design of $\mathbf{x}$ and the third question about shape normalization and alignment to the proceeding chapters, where we concentrate on specific object recognition applications. In this chapter, we will give brief descriptions of the subspace techniques in general terms.

The formal definition of the subspace is as follows: A subspace is a subset S of $\mathbb{R}^N$ with the following properties:

- The zero vector $\mathbf{0}$ is an element of S.
- If $\mathbf{u}, \mathbf{v} \in S$, then $\mathbf{u} + \mathbf{v} \in S$.
- If $\mathbf{u} \in S$ and $c \in \mathbb{R}$, then $c\mathbf{u} \in S$.

These properties make the subspace closed under addition and multiplication. Therefore, any linear combination of vectors in the subspace is still in the subspace. In this thesis, we will deal with such linear subspaces. We will assume the following linear model: $\bar{\mathbf{x}} \approx \mathbf{\Phi b}$, where $\mathbf{x}$ contains data about the shape of an object, $\mathbf{\Phi}$ is the matrix of basis vectors and $\mathbf{b}$ is the vector of new coordinates. Expressing the original observations in terms of a subspace basis $\mathbf{\Phi}$ means a change of coordinate system, different from the one in which the original data vector $\mathbf{x}$ is represented. The two questions regarding this model are as follows: How should we obtain the representation vector $\mathbf{x}$, and how should we construct the set of basis vectors that form the columns of $\mathbf{\Phi}$?

The relevant information of the observations is supposed to be expressed in terms of basis vectors spanning the subspace and any irrelevant information is left in the complement of the subspace. The criterion for the "relevant information" depends on the application. The criterion may be the fidelity to the source of information, so that any noise not coming from this source is suppressed. Or it may be the "visual quality" in the case of DCT-based image coding. In our case, we would like the relevant information to be the essence of the shape that determines

the object's class identity.

The subspace techniques can be categorized with respect to the construction of the basis $\mathbf{\Phi}$. The basis of the subspace can be model-driven or data-driven. In the model-driven case the basis vectors are fixed, such as dirac functions, cosines, complex exponentials, wavelets, and even random basis vectors. They are usually chosen because they have particular desired properties. In the case of Fourier analysis, the complex exponentials have the property of being the eigenfunctions of linear shift-invariant systems. They are useful to measure the frequency content of a signal. Likewise, wavelets provide good compromise between localized features of a signal and its frequency content.

In the data-driven methods, the bases are recovered from a set of observations with respect to some criteria: Uncorrelatedness (PCA), independence (ICA), non-negativity (NMF), classification accuracy (LDA), sparsity, etc. These techniques capture the structure or the distribution of the data in the original space. In most cases, observations of interest of many computer vision applications do have much less degree of freedom than the original representation frame suggests. In other terms, the intrinsic dimension of observations is less than the dimension of the original space. The data-driven methods are particularly beneficial in these cases.

The data-driven methods assume the availability of a set of observations, namely a training set. Let this set be composed of D observations (or instances, realizations, objects) from an N-dimensional space: $\{\mathbf{x}_1,\ \mathbf{x}_2,\ ..., \mathbf{x}_D\}$. The data matrix is an $N \times D$ matrix, where each observation is placed into a column. The statistics of the shapes will be determined by this data matrix, hence its construction is a crucial step. The desired properties of the data matrix can be listed as follows:

- There should be good correspondence among observations. The value at a particular index of the data vector $\mathbf{x}$ (or a particular variable) should correspond to the same measurement among all the vectors of the data matrix.
- The samples in the data matrix should represent the population well enough.

Unseen structures that exhibit statistical properties different than the training observations will not be adequately modeled by the subspace method. Most of the essential information of this unseen structures may not be expressed in terms of basis vectors.

- It is desired to have fewer variables than the number of samples. Otherwise, we will have fewer samples than the dimensionality of the original space, a situation which is referred to as the curse of dimensionality [5]. However, this is the case for many applications. For example, in biometry, it is difficult to collect data from many subjects, however there is abundant information (or measurements) per subject: The high resolution scans of 3D faces or high resolution hand scans. Fortunately, the measurements are highly correlated and we can assume that the samples are populated in a low dimensional subspace.
- There should be enough samples from each class. This requirement is especially important for supervised data-driven subspace techniques such as Linear Discriminant Analysis (LDA).

Another categorization of subspace techniques is based on the use of the class information of the observed data. A subspace can be unsupervised (generative, reconstructive) or supervised (discriminative). In the generative approach, the objective is to reduce the mean square error between the original data and the data projected onto the subspace under some pre-determined constraints. A well-known example of the generative approach is the principal component analysis (PCA). DFT, DCT and wavelet-based subspace approaches also fall in this category. In the case of pattern recognition, there are two main reasons for the use of generative models: (i) Their ability to greatly reduce the dimensionality of the data, and (ii) The hope that the essential structures related to class variations are expressed in terms of the coordinates of the subspace in use [5].

In the discriminative approach, the class information of the training samples is utilized to extract the basis vectors of the underlying subspace. Linear discriminant analysis (LDA) is a classical example, where the aim is to build a subspace spanned by the vectors that best discriminate among classes [6].

In the following sections, we will give descriptions of the particular subspace techniques we have used in our applications.

2.1. Direct Comparisons

The columns of the identity matrix **I** of size $M \times M$ provide a complete orthogonal basis for vectors of dimension M. These vectors are called Dirac basis. The projection on them is trivial and provides an extremely localized representation of the signal **x**.

The use of Dirac basis, or what we call the direct comparison method, simply suggests to stay in the original representation space and use the coordinates of whatever frame is given. The distance between two objects is just the L_1 or L_2 distance of the corresponding points in the original space. However, this original space is usually very high dimensional, making it ineffective for most object recognition applications.

2.2. Discrete Fourier Transform (DFT)

The DFT transforms a signal from time or spatial domain to the frequency domain. The basis of the space is now defined in terms of complex exponentials and the objects are placed in this space as points with respect to their frequency contents. A filtering operation to trim out some frequency axes (for example low pass filtering) results in representations of the signals in a subspace of the complete frequency space. If the signal is varying smoothly in the spatial domain, Fourier basis provides parsimonious representations. DFT-based subspaces are model-driven, since the basis vectors, the sinusoids, are chosen independent of the data being modeled.

Following subsections give formal definitions of 1D, 2D and 3D DFT, which are used in this thesis study. The dimension of the DFT (1D, 2D and 3D) here has a different meaning than the dimensionality of the observation (number of real numbers used for the exact representation); it rather indicates the spatial organization

(or neighborhood) of the data points.

2.2.1. 1D-DFT

We will assume that the original data space is N dimensional, e.g. the length of vectors to be transformed to the frequency domain is N. Consider the following matrix $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} w_N^{0\cdot 0} & w_N^{0\cdot 1} & \cdots & w_N^{0\cdot(N-1)} \\ w_N^{1\cdot 0} & w_N^{1\cdot 1} & \cdots & w_N^{1\cdot(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_N^{(N-1)\cdot 0} & w_N^{(N-1)\cdot 1} & \cdots & w_N^{(N-1)\cdot(N-1)} \end{bmatrix} \tag{2.1}$$

where $w_N = \exp\{2\pi i/N\}$ is a primitive N'th root of unity. The columns of the matrix $\mathbf{F}$ correspond to the harmonics that form the basis for the N-dimensional space of the frequency domain. Then a vector in the original domain $\mathbf{x}$ can be transformed into the frequency domain by $\mathbf{X} = \mathbf{F}^H\mathbf{x}$. This definition of DFT corresponds to the 1D-DFT, where the basis of the frequency space is composed of one dimensional harmonics.

2.2.2. 2D-DFT

Consider a 2D matrix $\mathbf{x}$ of size $N_1 \times N_2$. The 2D-DFT of $\mathbf{x}$ is a 2D matrix $\mathbf{X}$ of size $N_1 \times N_2$ and its elements are calculated as:

$$\mathbf{X}_{k_1k_2} = \sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} w_{N_2}^{-n_2\cdot k_2} w_{N_1}^{-n_1\cdot k_1}\mathbf{x}_{k_1k_2} \tag{2.2}$$

for $k_1 = 0,\ 1,\ ...,\ N_1 - 1$, $k_2 = 0,\ 1,\ ...,\ N_2 - 1$, and where $w_{N_1} = \exp\{2\pi i/N_1\}$ and $w_{N_2} = \exp\{2\pi i/N_2\}$.

The basis of the $N_1 \times N_2$ dimensional frequency space is constructed from the

$N_1 \times N_2$ matrices of the form $\mathbf{F}_{k_1 k_2}$:

$$\mathbf{F}_{k_1 k_2} = \begin{bmatrix} w_{N_1}^{k_1 \cdot 0} \\ w_{N_1}^{k_1 \cdot 1} \\ \vdots \\ w_{N_1}^{k_1 \cdot (N_1 - 1)} \end{bmatrix} \begin{bmatrix} w_{N_2}^{k_2 \cdot 0} & w_{N_2}^{k_2 \cdot 1} & \cdots & w_{N_2}^{k_2 \cdot (N_2 - 1)} \end{bmatrix} \tag{2.3}$$

2.2.3. 3D-DFT

Let $\mathbf{x}$ be a 3D-array of size $N_1 \times N_2 \times N_3$. Its 3D-DFT is a complex 3D-array of the same size and its element at the index (k_1, k_2, k_3) is calculated as:

$$\mathbf{X}_{k_1 k_2 k_3} = \sum_{n_3=0}^{N_3-1} \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} w_{N_3}^{-n_3 \cdot k_3} w_{N_2}^{-n_2 \cdot k_2} w_{N_1}^{-n_1 \cdot k_1} \mathbf{x}_{k_1 k_2 k_3} \tag{2.4}$$

for $k_1 = 0,\ 1,\ ...,\ N_1 - 1$, $k_2 = 0,\ 1,\ ...,\ N_2 - 1$, $k_3 = 0,\ 1,\ ...,\ N_3 - 1$, and where $w_{N_1} = \exp\{2\pi i / N_1\}$, $w_{N_2} = \exp\{2\pi i / N_2\}$, $w_{N_3} = \exp\{2\pi i / N_3\}$.

2.3. Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) is similar to DFT in the sense that it transforms a vector into a space where the basis vectors are harmonic signals. While in DFT the harmonic signals are complex exponentials oscillating in different frequencies, in DCT they are real-valued cosine signals. The DCT of a vector $\mathbf{x}$ of length N is

$$\mathbf{X}_k = \sum_{n=0}^{N-1} \mathbf{x}_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \tag{2.5}$$

for $k = 0,\ 1,\ ...,\ N - 1$. This corresponds to the definition of 1D-DFT. The 2D-DFT of an $N_1 \times N_2$ matrix $\mathbf{x}$ is calculated as follows:

$$\mathbf{X}_{k_1 k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \mathbf{x}_{n_1 n_2} \cos\left[\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right] \tag{2.6}$$

for $k_1 = 0,\ 1,\ ...,\ N_1 - 1\ ,\quad k_2 = 0,\ 1,\ ...,\ N_2 - 1.$

DCT is widely used for feature extraction and compression, because it has a strong "energy compaction" property. If the signal is highly correlated and smoothly varying in the spatial domain, the DCT summarizes most of the information in few low frequency coefficients. This property makes DCT-based subspace a good choice for reconstructive purposes. Its decorrelation ability can approach that of Karhunen-Loève Transform (KLT), and it has the additional advantage of providing fixed basis functions (model-driven subspace) excluding the necessity of building data-driven basis through decorrelation of training data.

2.4. Angular Radial Transform (ART)

Angular radial transform (ART) is a complex transform defined on the unit disk. The basis functions $V_{nm}(\rho, \theta)$ are defined in polar coordinates as a product of two separable functions along the angular and radial directions:

$$V_{nm}(\rho, \theta) = A_m(\theta) R_n(\rho) \tag{2.7}$$

where

$$A_m(\theta) = \frac{1}{2\pi} \exp(jm\theta) \tag{2.8}$$

and,

$$R_n(\rho) = \begin{cases} 1 & n = 0 \\ 2\cos(\pi n \rho) & n \neq 0 \end{cases} \tag{2.9}$$

Figure 2.1 shows real parts of the ART basis functions. As can be observed from this figure, with increasing order n, the basis functions vary more rapidly in the radial direction, whereas the order m expresses the variation in the angular direction. The angular radial transform of an image $f(\rho, \theta)$ in polar coordinates is a set of ART

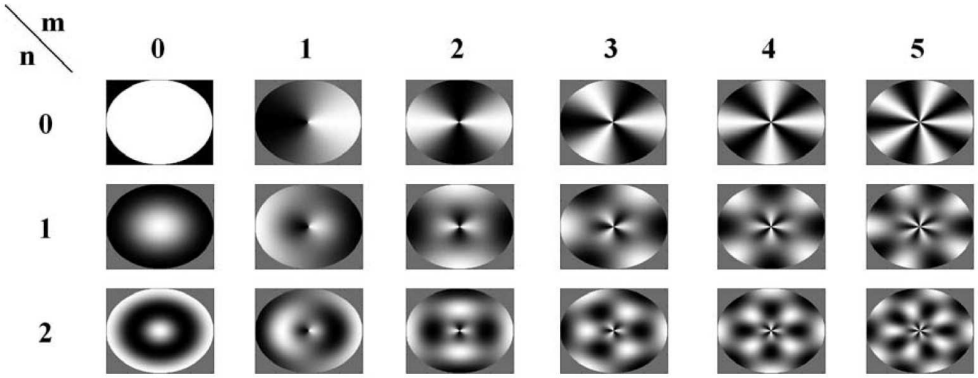


Figure 2.1. Real parts of ART basis functions.

coefficients $\{F_{nm}\}$ of order n and m. These ART coefficients can be derived as follows:

$$F_{nm} = \int_0^{2\pi} \int_0^1 V^*_{nm}(\rho, \theta) f(\rho, \theta)\, d\rho\, d\theta \tag{2.10}$$

A set of $N \times M$ ART magnitude coefficients can be used as features for recognition of images. In shape recognition, the ART coefficients are normalized to F_{00} in order to achieve scale invariance.

2.5. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is one of the most widely used feature extraction schemes in computer vision. The assumption is that the high-dimensional (N) representations of raw data structures are intrinsically low dimensional (K) and they lie on (K)-dimensional linear manifolds.

An active area of computer vision research that employs PCA, is the recognition of human faces from 2D intensity images [7, 8]. PCA is used to decouple the variations due to illumination and viewing direction and the variations due to identity. Another influential application of PCA is the recognition of general objects from their 2D intensity images. Murase and Nayar [9, 10] proposed a system which captured images of general objects with varying pose and illumination. Then these

images were reduced to a 20-dimensional subspace via PCA. Indeed it has been proved that the images of a Lambertian surface taken from arbitrary view directions and under various illumination conditions lie in close to a nine-dimensional subspace [11, 12]. In these two applications, the raw data are represented in pixel values organized in $M \times N$ image matrices. Other influential approaches that employ PCA are Active Shape Modeling [13] and Active Appearance Modeling [14]. The authors emphasize the importance of correspondence building among various instances of structures to be recognized and present a number of techniques to build correspondence (Procrustes analysis [15], thin-plate splines [16]).

PCA decorrelates the data using second order statistics. Reliance on the second order statistics is based on the assumption that the observations are Gaussian. For multivariate Gaussian data, the mean and covariance determines all the statistical behaviour. The axes of large variance are assumed to describe the underlying structure, while axes of small variance are considered as noise.

For a data matrix $\mathbf{X}$ and the mean vector of the data being μ, eigenvectors of the $M \times M$ covariance matrix, $\mathbf{C} = (\mathbf{X} - \mu)(\mathbf{X} - \mu)^T$ gives the principal directions of variations. Notice that the covariance matrix is equivalent to the correlation matrix of centered data.

Let $\{\mathbf{u}_1,\ \mathbf{u}_2,\ ...,\ \mathbf{u}_K\}$ be the first K eigenvectors of $\mathbf{C}$ with corresponding eigenvalues $\{\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_K\}$. These vectors model the largest variations among the training samples, therefore they are considered to capture most of the significant information. The amount of information maintained depends on K and the spread of eigenvalues. The projection of an input vector $\mathbf{x}$ onto the PCA subspace is given by $\mathbf{a} = \mathbf{U}^T\mathbf{x}$, where $\mathbf{U}$ represents the $M \times K$ projection matrix formed as $[\mathbf{u}_1\ \mathbf{u}_2\ ...\ \mathbf{u}_K]$.

Apparently, PCA is a reconstructive and data-driven approach. It is the best linear dimension reduction method in terms of the mean-square error.

2.6. Independent Component Analysis (ICA)

ICA has been successfully used in many different applications for finding hidden factors within data to be analyzed or decomposing it into the original source signals, namely the blind source separation problem. In the context of natural images, it also serves as a useful tool for feature extraction [17] and person authentication tasks [18, 19].

ICA is a generalization of PCA in that it removes correlations of higher order statistics from the data. With ICA, we assume that the observed signals $\{\mathbf{x_1},\ \mathbf{x_2},\ ...,\ \mathbf{x_N}\}$ result from linear mixtures of K source signals $\{\mathbf{s_1},\ \mathbf{s_2},\ ...,\ \mathbf{s_K}\}$. Let the dimensions of the observed signals and the source signals be the same and equal to K. We admit the signal model, $\mathbf{X} = \mathbf{AS}$ where $\mathbf{A}$ is the $K \times K$ matrix of mixing coefficients and $\mathbf{S}$ contains source signals in its rows. Both the source signals and the mixing coefficients are unknown, and need to be estimated. Our aim is to find a linear transformation, $\mathbf{W}$ such that $\mathbf{Y} = \hat{\mathbf{S}} = \mathbf{WX}$, where $\mathbf{W}$ is the separating or de-mixing matrix.

2.6.1. The FastICA Algorithm

The objective is to separate the input vectors into statistically independent sources. Denote the jth element in the random vector $\mathbf{y}$ as y^j and assume that these elements are random variables. If these random variables are independent, the probability distribution function of the random vector $\mathbf{y}$ is

$$f_{\mathbf{y}}(\mathbf{y}) = f(y^1,\ y^2,\ ...,\ y^K) = \prod_{j=1}^{K} f_{y^j}(y^j) \tag{2.11}$$

We want to find the matrix $\mathbf{W}$, such that $\mathbf{y} = \hat{\mathbf{s}} = \mathbf{Wx} = \mathbf{WAs}$ is satisfied and the $\{y^j\}$ are mutually independent. A way to maximize the independence condition is to define a function whose global optima coincide with the case of the independence of the variables. Then iterative optimization methods are used to find one of those optima. We have chosen the FastICA algorithm proposed by Hyvärinen and Oja

[20] and used the FastICA throughout our implementations in this thesis. In the following, we mainly reproduce the formulation introduced in [20].

In the FastICA algorithm [20], this function is defined in relation with the negentropy J of a random vector $\mathbf{y}$:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{2.12}$$

where $\mathbf{y}_{gauss}$ is a Gaussian random vector with the same covariance matrix as $\mathbf{y}$. $H(\mathbf{y})$ is the differential entropy of the random vector y and is equal to $-\int f_{\mathbf{y}}(\mathbf{y}) \log f_{\mathbf{y}}(\mathbf{y}) dy$. The negentropy can be viewed as a measure of non-Gaussianity of the random vector $\mathbf{y}$. If the random variables $\{y^j\}$ are uncorrelated, then we can relate the negentropy to the mutual information of $\{y^j\}$ as follows:

$$I(y^1,\ y^2,\ ...,\ y^K) = J(\mathbf{y}) - \sum_j J(y^i) \tag{2.13}$$

The mutual information measures the dependence of random variables. The FastICA algorithm aims to minimize the mutual information of the components $\{y^j\}$, with respect to $\mathbf{W}$.

The negentropy can be approximated as follows:

$$J(y^j)) \approx c[E\{G(y^j)\} - E\{G(v)\}]^2 \tag{2.14}$$

Here, G is a non-quadratic function, v is a Gaussian random variable with zero mean and unit variance and c is any positive constant. Let $\mathbf{w_j^T}$ be the jth row vector of $\mathbf{W}$. To find one independent component, the following function is maximized with respect to $\mathbf{w_j}$:

$$J_G(\mathbf{w_j^T}) = c[E\{G(\mathbf{w_j^T x})\} - E\{G(v)\}]^2 \tag{2.15}$$

Recall that the mutual information is minimized when the sum of the negentropies of $\{y^j\}$ is maximized. The following optimization problem is solved to maximize the independence criterion: Maximize $\sum_{j=1}^{K} J_G(\mathbf{w_j^T})$ with respect to $\mathbf{w_j}$, $j = 1, 2, ..., K$, under constraint $E\{(\mathbf{w_m^T x})(\mathbf{w_n^T x})\} = \delta_{mn}$.

Usually, the non-quadratic function $G(y)$ is selected as y^4. The optimization problem is solved with a fixed-point algorithm described in [20].

2.6.2. ICA1 and ICA2 Architectures

There are two different interpretations of the source-mixing assumption, which are denoted as ICA architecture I (ICA1) and ICA architecture II (ICA2) [20]. In ICA1, observations are considered to be a mixture of statistically independent sources, i.e., basis signals; however the estimated mixture coefficients are not statistically independent. In ICA2, the mixture coefficients should be estimated under the assumption of independence, whereas the basis signals are not independent.

ICA2 is similar to PCA in the sense that it provides global features. The basis vectors are not sparse; so the ICA2 coefficients are influenced by every point of the input raw data x. On the other hand, ICA1 is similar to NMF. The basis vectors are sparse, hence the ICA1 coefficients reflect localized activity. The choice of the architecture (ICA1 or ICA2) depends on the nature of the application [18]. Draper et al. [18] argue that the task of facial identity recognition is holistic and is better handled by global feature vectors; whereas the localized feature vectors are more suitable to facial action recognition. Therefore, they claim that ICA2 architecture is preferable to identity recognition. This argument is verified in [21], where ICA2 outperformed ICA1 for the task of person identification via global hand shape and appearance. If the object to be recognized is complete (no occlusion or missing data), we expect better performance from holistic approaches.

Throughout the thesis, we plug ICA2 architecture to obtain holistic descriptions of the objects of consideration. For a parts-based analysis we will rather employ

NMF.

Prior to the estimation of the de-mixing matrix, W, it is conventional to reduce the dimensionality of the data matrix via PCA. The columns of the new data matrix are constituted of the projections of the training samples onto K-dimensional subspace obtained by PCA.

2.7. Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization is another matrix factorization technique with the added constraint that each factor matrix have only nonnegative coefficients [22]. It has been observed that avoiding the artificiality of negative coefficients enhances physical significance of the component sources. In fact, each source resembles a part of the object leading to a parts-based description. A case is the NMF decomposition of 2D intensity faces, where the basis vectors are found to reflect the local features of faces.

Given a nonnegative data matrix, $\mathbf{X}$, of size $M \times N$, we factorize it into two nonnegative matrices $\mathbf{V}$ and $\mathbf{H}$, such that $\mathbf{X} \approx \mathbf{VH}$, with sizes $M \times K$ and $K \times N$, respectively. $\mathbf{V}$ contains the basis vectors in its columns and $\mathbf{H}$ is constituted of combination coefficients.

We use the multiplicative update rules described by Lee and Seung [23] to estimate the nonnegative $v_{m,k}$ and $h_{k,n}$ factors. The objective function is taken as $||\mathbf{X} - \mathbf{VH}||^2$, where $||\cdot||$ is the Frobenius norm and the factor matrices are constrained to have nonnegative elements [23]. They first define additive update rules, based on the gradient descent over an objective function that optimizes $\mathbf{V}$ and $\mathbf{H}$. Then, by selecting appropriate step sizes, they convert the additive update rules into multiplicative ones.

Setting the objective function that measures the reconstruction error as the

square of the Euclidean distance between $\mathbf{X}$ and $\mathbf{VH}$ as

$$[\mathbf{X} - \mathbf{VH}]^2 = \sum_{ij} \left((\mathbf{X})_{ij} - (\mathbf{VH})_{ij} \right)^2 \tag{2.16}$$

the optimization problem is to minimize $\|\mathbf{X} - \mathbf{VH}\|^2$ with respect to $\mathbf{V}$ and $\mathbf{H}$, subject to the constraints $\mathbf{V}, \mathbf{H} \geq 0$. The multiplicative update rules [23] solving this problem are as follows:

$$\mathbf{H}_{a\tau} \leftarrow \mathbf{H}_{a\tau} \frac{(\mathbf{V}^\mathbf{T}\mathbf{X})_{a\tau}}{(\mathbf{V}^\mathbf{T}\mathbf{V}\mathbf{H})_{a\tau}} \quad , \quad \mathbf{V}_{ia} \leftarrow \mathbf{V}_{ia} \frac{(\mathbf{X}\mathbf{H}^\mathbf{T})_{ia}}{(\mathbf{V}\mathbf{H}\mathbf{H}^\mathbf{T})_{ia}} \tag{2.17}$$

Notice that in PCA and ICA, both basis vectors and coefficients can have positive and negative values, and the reconstruction may therefore involve cancellations of irrelevant parts. This introduces unphysical artifacts of negative mass or luminance. Since only positive bases and coefficients are involved in NMF, that is, subtractions are not allowed in linear combinations, NMF leads to basis signals that are locally physical and that model partial structures of objects [22].

3. HAND BIOMETRY

This chapter provides a survey of hand biometric techniques in the literature and incorporates several novel results of hand-based person identification and verification. We compare several feature sets in the shape-only and shape-plus-texture categories and we emphasize the relevance of a proper hand normalization scheme in the success of any biometric scheme. The preference of the left and right hands or of ambidextrous access control is explored. Since the business case of a biometric device partly hinges on the longevity of its features and generalization ability of its database, we have tested our scheme with time lapse data as well as with subjects that were unseen during the training stage. Our experiments were conducted on a hand database that is an order of magnitude larger than any existing one in the literature.

3.1. Introduction

Hand recognition systems are among the oldest biometric tools for automatic person authentication. Access control devices have been manufactured and commercialized since the late seventies. Several patents have already been issued for hand recognition devices [24, 25, 26, 27, 28] and live applications have been launched and used at nuclear plants, airports, hotels in the last 30 years [29, 30]. The first biometric device was manufactured in 1971, and it was indeed a hand-based recognition tool called Identimat [24]. Hundreds of Identimat devices were used for security purposes at the Department of Energy, U.S. Naval Intelligence in the 1970s. However hand biometry has gained interest in the academic circles, mostly with the progress of computer vision research, only in the last decade.

Hand-based person recognition provides a reliable, low-cost and user-friendly, all in all, a viable solution for a range of access control applications. Other "nearest competitor" modalities are face, iris, fingerprint and retinal biometry. The face recognition alternative is another low-cost solution for access control. However,

unless several challenging issues are satisfactorily solved, such as illumination, pose and facial expression variations and occlusions due to accessories, it will be limited to controlled niche applications. Unsupervised face recognition, where the user does not have to pose for the camera, requires both detection and segmentation of the facial region from cluttered backgrounds and normalization of the face, both challenging problems. Despite its attraction, automatic face recognition within its current state of the art is regarded as a biometric modality with inadequate reliability.

The iris and retinal modalities demand specialized acquisition devices. Furthermore due to their intrusive nature, most people feel uncomfortable and they will not, in all likelihood, be widely deployed. Fingerprint modality is by far the most studied case, commonly used from forensic evidence collection to personal device access, home access or Internet-access. However, minutiae are very sensitive to cuts and wounds in the finger, hence fingerprint features from manual laborers or elderly people become less reliable and more difficult to acquire. In fact, up to four per cent of the population may fail to provide fingerprints with acceptable quality [31]. Most people have still a certain reticence with fingerprints; for example, fingerprints are considered to be private by some users and they may not yield fingerprint for commercial applications. There has also been a considerable amount of research on voice authentication, especially in telephone applications. However speech data suffer from intrapersonal variations due to mood, emotions, illnesses and ageing. Due to these handicaps, its reliability is low and voice-based authentication is not yet a competitor to fingerprint or hand.

In contrast to these techniques hand biometry offers some advantages. First, data acquisition is economical via commercial low-resolution scanners or cameras, and its processing is relatively simple. Second, according to two public surveys [32, 33] people like hand-based access systems, they do not consider hand information to be as private as iris or fingerprint in daily applications, hence they find it less invasive and more convenient to use than other biometric modalities. Third, hand-based access systems are very suitable for indoor and outdoor usage, and can work well in extreme weather and illumination conditions [32, 33]. Fourth, hand features of

adults are more stable over time and are not susceptible to major changes, except for injuries- or arthritis-based deformations. Finally, hand-based biometric information has been shown to be very reliable and can successfully recognize people among populations of the order of several hundreds [21, 34, 35]. We conjecture, therefore, that time has come to deploy hand biometric devices for daily applications ranging from access to hospitals, child daycare centers, industrial plants, sport centers and libraries of universities to more challenging situations at border control and airports. It can also be used to enhance the security of e-commerce and banking applications via integration to the conventional systems using PIN codes and passwords.

In this chapter we provide an extensive literature survey on hand biometry and present performance results with a wide variety of subspace-based methods. We emphasize the importance of hand normalization as a crucial pre-processing step of our methods. We consider the generalization ability of the ICA-based feature extraction algorithm from small to large populations, the preference for right or left hand, the advantage of ambidextrous testing, the performance of new features, and various fusion schemes to improve the performance. In addition to the analysis of our global hand appearance based approach, we provide comparative performance results of various techniques on a large database. Some of these techniques were previously applied on hands for person recognition; others are considered first as tools of characterizing human hands for biometric purposes, such as Principal Component Analysis of global hand appearance, Active Appearance model, Fourier descriptors of hand contour and Angular Radial Transform.

3.2. Characteristics of the Human Hand

In this section, we describe the characteristics of the human hand and its relevant aspect for feature extraction.

3.2.1. The Skeleton of the Hand

The anatomic structure and biomechanics of the human hand have interested researchers working in the areas of computer animation, hand gesture and sign language recognition. This information is also beneficial for hand biometry.

The hand contains 27 bones, categorized into three groups: The carpals in the wrist, the metacarpal bones that run along the palm, and the phalanx bones in the fingers [36]. Figure 3.1 shows the skeletal model of the human hand. When laid on a flat surface, the interphalangeal joints at the fingers become fixed since the extension/flexion of the fingers are disabled. However, those of the thumb can still move slightly since they are not totally in the supine position. The carpal-metacarpal joints are already limited in their freedom of movement, again except for the thumb. Thus, a hand lying on a flat surface is reduced to seven degrees of freedom, three at the three joints of the thumb, and four at metacarpal-phalanx joints of the four fingers. The metacarpal-phalanx joints (MCP) are the pivots where fingers make adduction/abduction movement, i.e. lateral movements on the plane. The orientation of the thumb, on the other hand, is controlled by its carpal-metacarpal joint (TM in Figure 3.1) and the thumb shows relatively high in-plane flexibility.

Kuch and Huang [37] used a set of constraints on finger movements for gesture modeling where the range of in-plane rotation angles of the four fingers around their pivot (MCP joints) is taken between -15 and 15 degrees. A more complex set of relations were assumed between the in-plane angles of the three joints of the thumb. Lin et al. [38] developed another hand-skeleton model under similar assumptions, with the additional constraint of a rigid middle finger. In our hand-posture normalization scheme [21, 34] we make use of the five degrees-of-freedom model, so that we rotate the fingers to preset reference angles based on an estimate of their metacarpal pivot locations [34]. The posture normalization algorithm is described briefly in Section 3.4.2 and in more detail in our paper [34].

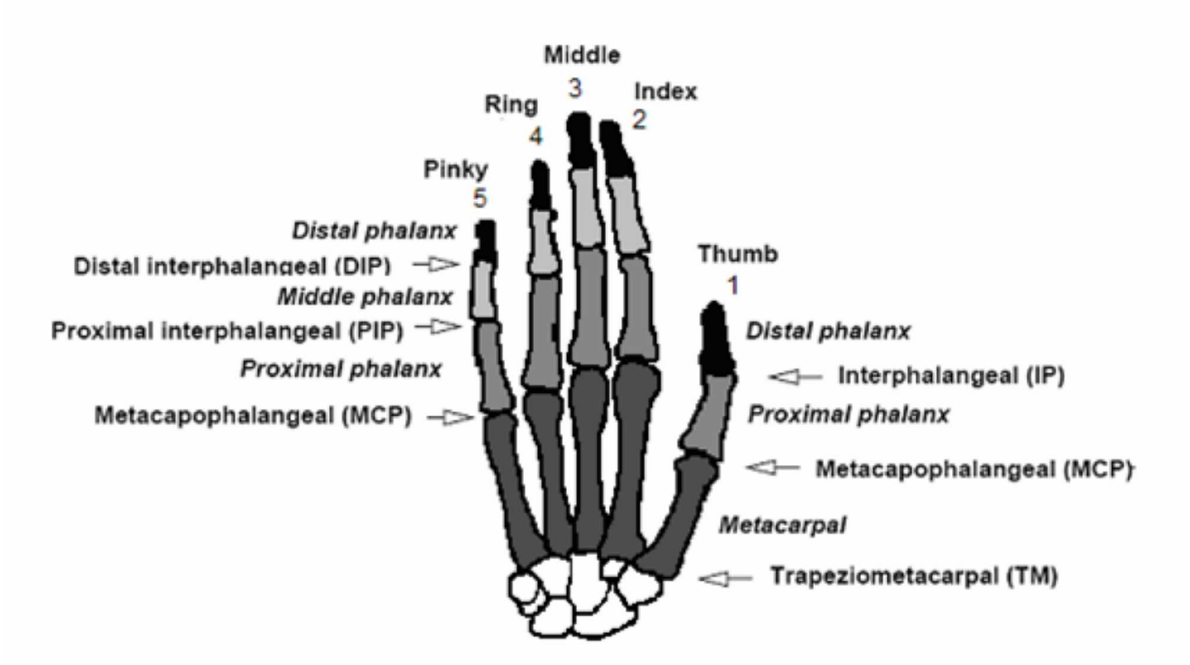


Figure 3.1. The skeleton of the hand.

3.2.2. The Geometry of the Hand

Geometrical measures have been used in most of the patented methods of hand-based identification and in earlier publications. Ernst [25] mentioned the anthropological studies where it was stated that the length and breadth of the hand had very little statistical correlation. Since both sizes were useful measures, he developed a string-based, mechanical aperture to measure the width and the length of the hand. Miller [24] advanced this scheme with an electro-mechanical system, called Identimation, which measured the lengths of the four fingers and compared them with measurements prerecorded on an identification card. In 1972, Jacoby et al. [26] came up with the first optical system that measured the distances between finger tips and finger crotches through a scanner.

Geometrical features of the hand, referred to also as ”hand dimensions” in the literature, constitute the bulk of hand features adopted in most hand recognition systems. One advantage is that geometrical features are more or less invariant to

the global positioning of the hand and to the individual planar orientations of the fingers. Among numerous geometrical measures we can cite lengths, widths, areas, and perimeters of the hand, fingers and the palm. Jain et al. [39], have come to the conclusion that hand geometrical features solely are not sufficiently discriminative. This is due to the fact that they are somewhat correlated and there are at most 50 geometrical features. For the present state of the art, they are not viewed as suitable for identification (one-to-many comparison) purposes, but instead can be used for verification (one-to-one comparison) tasks [39]. Therefore, for more demanding applications one must revert to alternative features such as hand global shape, appearance and/or texture.

Hand geometrical features consist of a set of measured dimensions, such as lengths, widths and areas of the fingers, of the hand and of the palm. Jain et al. [39] use 16 axes predetermined with the aid of five pegs. The gray-level profiles along these axes are modeled as an ideal profile contaminated by Gaussian noise. Using this profile model, 15 geometrical features are extracted and tested for verification. In their peg-aided identification system, Sanchez-Reillo et al. [40] use a similar set of geometric features, containing the widths of the four fingers measured at different latitude, the lengths of the three fingers and the palm. The distances among three interfinger points (finger valleys) and the angles between the lines connecting these points are also part of the set. Wong and Shi [41], in addition to finger widths, lengths and interfinger baselines, employ the fingertip regions. The fingertip regions correspond to the top one-eighth portion of the index, middle and ring fingers. The curves extracted from these fingertip regions are then aligned, resampled and compared via the Euclidean distance. Bulatov et al. [42] describe a peg-free system where 30 geometrical measures are extracted from the hand images. In addition to widths, perimeters and areas of the fingers, they also incorporate the radii of inscribing circles of the fingers and the radius of the largest inscribing circle of the palm. They, however, do not give any information on the extraction procedure of these features.

While geometrical features are simple to extract they have certain disadvan-

tages. First, they are not discriminating enough to be used in identification tasks and in high-security verification scenarios. The reason is that this approach reduces the holistic shape information to a small set of features, and obviously texture cannot be exploited. Furthermore, a simple set of geometrical measures can be more easily faked or compromised. For these reasons, some authors propose the fusion of geometry-based features with other characteristics of the hand such as the finger shapes [43] or the palmprint features [44, 45, 46].

3.2.3. The Shape of the Hand

The shape or the silhouette of hand has gained little attention in the literature for person identification despite considerable literature on shape matching in computer vision. Jain and Duta [47] were the first to propose deformable shape analysis, and to develop an algorithm where hand silhouettes are registered and compared in terms of the mean alignment error.

The hand shape, surprisingly, exhibits great variation among individuals. The silhouettes contain much richer information as compared to geometrical measures of the hand. For example, the roundness of finger tips, the shape of the thumb, sharpness of finger valleys etc. are not necessarily incorporated in the geometric measurements. The geometrical features, no matter how much detailed, are surpassed by the shape features in parts-based or holistic analysis.

The major roadblock for the use of hand shape as a person identifier has been the fact that hand is a highly deformable and articulated organ, making it challenging to characterize the global shape. The intrapersonal variability of the hand shape, if not properly normalized, can be much bigger than the interpersonal differences. Thus, researchers often use pegs to fix the position of the hand and the orientation of the fingers [47].

3.2.4. The Palm of the Hand

Perhaps inspired by the recent advances in fingerprint analysis, the palm has attracted a lot of attention in the last decade. The palm exhibits a rich pattern of striations that are believed to be unique to each individual. In fact, palmprints have been utilized as person identifier for more than 100 years. These techniques, however, were not automated and required the application of ink, powders or other chemicals to put ridges into evidence. Notice that the ridgeology practice encompasses not only palms but also footprints and any other striated surface [48]. The use of palmprint features for computer-based identification was first proposed by Shu and Zhang [49] in 1998. Afterwards, D. Zhang and his colleagues have developed a series of computer vision algorithms for processing palmprint features.

The palmprint features can be divided into three categories based on their scale: (i) Palm lines including the principal lines, (ii) Creases or wrinkles, (iii) Ridges or the minutiae. The palm lines and the principal palm lines are discriminating features that are considered to be stable over time [50]. Creases or wrinkles are irregular lines that are thinner than the principal lines and ridges correspond to regular and very thin lines that are similar to the minutiae of the fingerprints. The extraction of the minutiae requires high-resolution imaging and elimination of palm lines and creases. The minutiae of the palmprint are as reliable for identification as those of the fingerprint, and have been used for forensic applications [48].

Shu and Zhang [49, 50] were the first to publish on palmprint-based person recognition. They applied nonlinear filters to detect the principal palm lines and encoded the detected lines by their end-points and mid-points. Duta et al. [51] used a set of feature points along the prominent palm lines and the associated line orientations to match two palmprint images. They did not explicitly extract palm lines as Shu and Zhang did [50], but used only isolated points along palm lines. Wu et al. [52] proposed a two-stage palm line extraction scheme. In the first stage, morphological operators are applied to the palm image to extract palm lines in different directions. In the second stage, a recursive process is used to trace and

complete the palm line using the local information along the regions extracted in the first stage. You et al. [53] proposed a hierarchical palm matching algorithm where global texture energy obtained by Laws' convolution masks [54] were used to select a small number of candidate palms at coarse level. An interest point-based matching algorithm was applied to the candidate palms at fine level to achieve the final decision. The interest points along the palm lines are similar to the feature points of Duta et al. [51] and are detected by local operators.

Palmprints have a large number of creases which are assumed to be stable in a person's life. In their work, Chen et al. [55] tried to detect the creases by using a direction computing method based on the local gray level values. Funada et al. [56] suggested the use of ridges for palmprint characterization. The ridge patterns, such as the termination of bifurcations, i.e. minutiae, are inherited from the fingerprint literature. However, the palmprint minutiae are crossed by many creases. Funada et al. [56] set out to first eliminate these creases and then extract ridge candidates by fitting the local image to a ridge model. A ridge pattern is approximated by a two-dimensional sine wave and the pairs of peaks are detected in the power spectrum of the local image.

In general, the palmprint features, such as principal lines, creases, wrinkles, delta points, minutiae, etc. are difficult to extract and characterize, especially in low resolution images. Researchers often used ink to enhance these line structures of the palm [49, 50, 53, 55]. Alternatively, instead of explicitly extracting and coding the palm lines, creases and interest points, edge maps can be used directly to compare palm images. The edge maps provide global information, even at low resolutions about the magnitudes and directions of the palm lines and creases. Wu et al. [57] used fuzzy directional element energy feature which provides line structural information about palmprints via encoding the directions and energies of the edges. Wu et al. [58] proposed a similar notion in one of their recent papers where they used directional line detectors to obtain a set of line magnitude images. Then these directional images were divided into overlapping grids and directional line energy features were computed. Han et al. [59] applied Sobel and morphological operators to the

central part of the palm image and used the mean values of the grid cells as features. Similarly, Kumar et al. [44, 45] have used line detection operators consisting of four-orientation convolution masks. The output of these operators are merged in one single directional map and standard deviation of pixels of overlapping blocks on the directional map are used as the palmprint features.

Li et al. [60] proposed the use of Hausdorff distance to compare the line edge maps of two palm images. The lines and curves, forming an edge map, are compared by Line Segment Hausdorff distance and Curve Segment Hausdorff distance.

An alternative way is to consider the central part of the palm as a textured image and apply well-known pattern recognition techniques to represent the palm region. These techniques include Gabor filters [61, 62, 63, 64], Global texture energy [53], Fourier transform [46, 65, 66, 67], Eigen palms through Karhunen-Loève transform [67, 68, 69, 70], Fishers' linear discriminant [69, 70], Zernike Moment invariants [71], Wavelets [69, 72, 73, 74], Independent Component Analysis [69, 75, 76], Correlation filter classifier [77], Haar wavelets [78], Global and local texture energy [79, 80] and Hu moment invariants [81].

3.2.5. The Fingers

Since the shape of the hand is characterized by great intra-person variation due to the articulation of fingers, some authors segment the hand into its fingers [43, 82, 83] in order to separately model the shapes of the individual fingers.

Oden et al. [43] proposed to model the shape of each individual finger with implicit polynomial functions of the fourth degree. Then the Keren invariants [84] are extracted from the fitted polynomials to be used as features invariant to affine transformations. Xiong et al. [82] separated and identified multiple rigid fingers under Euclidean transformations. The fingers are aligned with the aid of an elliptical model and their similarity is measured on finger width observed at predefined nodes. Fouquier et al. [85] proposed a method based on the projection of finger boundaries

on the major axis of the fingers. They segment the fingers using finger tips and inter-finger valleys. They compute the histogram of the distances from the finger boundary to the major axis of each finger. The histograms, smoothed with a Gaussian kernel, constitute their feature vectors.

Inner side of the fingers is textured with creases, whose location and pattern differ from person to person. Joshi et al. [86] proposed to use the gray-level values of the finger images for person verification. They have defined a feature called the "wide line integrated profile", which is obtained by averaging the gray-level values over five mm wide bands. The distinct peaks in the line profile correspond to the creases of the fingers. Two profiles are then matched by choosing the maximum of the correlation values calculated in a range of shift values. Since this scheme necessitates precise localization and alignment of fingers, the authors use a special acquisition setup consisting of a mechanical guide and a micro switch to get an already aligned finger image from the user. The system acquires one finger at a time, a constraint that decreases the user-friendliness of the system especially if multiple fingers are to be matched. Ribaric and Fratric proposed an eigenfinger approach, which is then fused with either eigenpalms [87] or finger geometry [88]. They extract strip-like finger subimages and apply Karhunen-Loève transform in order to obtain eigenfingers. These eigenfingers encode the texture variation among the fingers of the database.

3.2.6. Joint Hand Shape and Texture Features

The palmprint and hand shape information provide independent biometric identity features, hence one can benefit from their joint use for person recognition. The integration of palmprint and shape is generally performed at feature level by using palmprint features and simple geometrical measures together or at score or decision level by constructing classifiers guided by palmprint and shape-based experts [45].

Kumar et al. [45] fused the palmprint features and geometrical measures both

at feature level and at score level. In order to characterize the palm, they have used line detection operators consisting of convolution masks, each of which is tuned to one of the four orientations. The output of these operators are merged in one single directional map and standard deviation of pixels of overlapping blocks on the directional map are used as the palmprint features. Eighteen geometrical measures such as widths and lengths of the fingers and the palm are estimated to represent the shape. The palmprint and geometrical features are concatenated to form a single feature vector representing the hand. In addition to feature level fusion scheme, these authors also propose fusion at score level, where individual matching scores for palmprint and hand geometry are combined using the max rule.

3.3. Hand Image Acquisition

3.3.1. Acquisition Devices

After the first electromechanical devices focused on geometric features [24, 25]; the development of optical and infrared imaging technology made it possible to process hand images with computer-vision tools. Handkey device is a prototypical commercial product of Schlage Recognition Systems. The device originates from the invention of Sidlauskas who patented his scanning device in 1988 [27, 30]. The user positions his/her right hand horizontally between a set of pins that restricts the orientation of the fingers. The image of the hand is acquired by a CCD camera from above and, with the help of a mirror, from the side.

Other research groups developed their acquisition setups mostly inspired from the invention of Sidlauskas [39, 40]. This setup is suitable for extraction of hand shape, but it does not enable palmprint acquisition.

For systems based on palmprints, the imaging quality is more important. In early work, researchers used ink to get a palmprint on the paper, which were then digitized [49, 50, 53, 55]. This laborious technique is only feasible for very specific applications such as criminal identification. High quality palm images demand

contactless design and good illumination. D. Zhang and his colleagues were the first to develop such an acquisition device [62, 65, 89]. This device includes a ring-shaped source providing white fluorescent light, a platform with pegs to guide the users, a CCD camera, lens, frame grabber and A/D converter. It is intended for civilian applications such as access systems and ATMs.

While the device developed by Zhang et al. [62, 65, 89] only acquired palm images, Kumar et al. [44, 45] have collected data with a setup that can jointly acquire hand shape and palm image. Their device, however, necessitates an uncomfortable positioning of the user's hand facing upwards. Furthermore due to the curved nature of the back of the hand, the placement is not unique and this causes some yaw distortion in the hand.

The choice between a camera and a scanner for joint hand and palmprint imaging is discussed by Wong et al [89]. The camera is advantageous both due its acquisition speed and because it enables a non-contact setup. The contact of the hand with the scanner surface causes deformation in the palmprint features depending on the pressure level; and the scanner surface should be regularly cleaned up.

Flatbed scanners, on the other hand, provide a viable alternative where the user can lay comfortably his/her hand, and the resulting image is high-quality with homogenous dark background and constant illumination. Notice that to achieve conditions similar to those of a scanner, the camera setup should be fixed and focused on the hand, there should be a flat surface for the user to place her hand, and in many cases special illumination is needed. For web-based access systems, e-commerce and e-banking applications special hand or palmprint acquisition devices may not be affordable in the home and office environments. Instead, the ubiquitous flatbed scanner is the most appropriate capture device. Many researchers worked with hand and/or palmprint images acquired by flatbed scanners due to its simplicity and ease for data collection [21, 34, 41, 42, 51, 59, 66, 69, 81, 87].

Early hand acquisition devices used pegs controlling the finger orientations,

thus intending to constrain degrees of freedom for hand articulation [39, 40, 47, 62, 74, 86]. Presently, peg usage for constraining the position of the fingers is considered to be inappropriate for two reasons: First, it decreases the comfort or user-friendliness of the device due to the training stage to learn proper placement. Second, for people with too small or too big hands they may cause stress deformations especially in the inter-finger valleys due to hard contacts. The new trend is definitely to design peg-free systems [21, 34, 41, 42, 43, 44, 45, 46, 87]. These unconstrained acquisition systems rely on posture-independent features or preprocess hand images for posture normalization.

3.3.2. Which Hand to Acquire?

It might seem that the choice between right and left hand would be inconsequential for hand biometry. For example, since the majority of people are right-handed, it would be a matter of convenience to design right-handed devices. However, some authors have observed a performance difference between right and left hands. For example, Kumar and Shen [66] and Kumar and Zhang [90] have reported that the performance differences are of the order of 0.5 to one per cent. We conjecture that the statistical difference between the right and left hands could be due to the fact the working hand, often the right one, is plumper and its palm gets deformed more easily with device contact. Similar observations were made over time lapse images: The intra variations of the right hand are comparatively more over time.

In many studies, the left and right hand palms of the same person were considered independently, hence as if belonging to different classes, and the performance measurements were done accordingly [57, 61, 62, 63, 64, 68, 69, 78, 79]. In fact, the palmprints and the geometry of the right and left hands of the same person are highly correlated, and the correlation between these two hands can be more advantageously exploited. In our previous work, we have shown that the intrapersonal feature distances between left and right hands were much smaller than the interpersonal distances between hands of different people [34].

One way to utilize the correlated information in the two hands is to apply fusion schemes. For example, Kumar and Zhang [90] used fusion of left and right palmprints with the sum rule at score level. In Section 3.6.5, we discuss various fusion schemes at data level, feature level, and score level.

3.4. Image Processing

In this section, we describe our novel hand normalization algorithm along with the discussion of the relevant work in the literature. When no positioning aids such as fixation pegs are used, hand images exhibit great intra-class variations due to hand placement (rotation and translation) and free finger orientations. With our normalization algorithm, we minimize posture variations and also correct for illumination variations due to the pressure of the hand on the scanner.

3.4.1. Segmentation of the Hand from the Background

For the hand placed on a platen of the acquisition device or on a scanner, the background is almost uniform and therefore segmentation becomes a relatively easy task. In some systems [39], hand segmentation is not even required, since the hand features are computed directly based on the peg template.

In the work of Jain and Duta [47], the mean-shift unsupervised segmentation and a contour following algorithm are used to extract the shape of the hand. In most other works, simple thresholding is used for segmentation [30, 41, 44, 87]. For example, Kumar et al. [44] have used Otsu's thresholding method. However, segmentation performed with simple thresholding is sensitive to many factors, such as accessories (rings, bracelets, watches) and sleeves, dirt artifacts and darker skin regions on the hand. The failure to correctly segment and extract the silhouette of the hand causes performance degradation as well as frequent rejection of the authorized users. Another important factor is the "portability" of the segmentation algorithm, i.e. the algorithm should be easily adapted to a new setup, with different imaging devices and environmental factors.

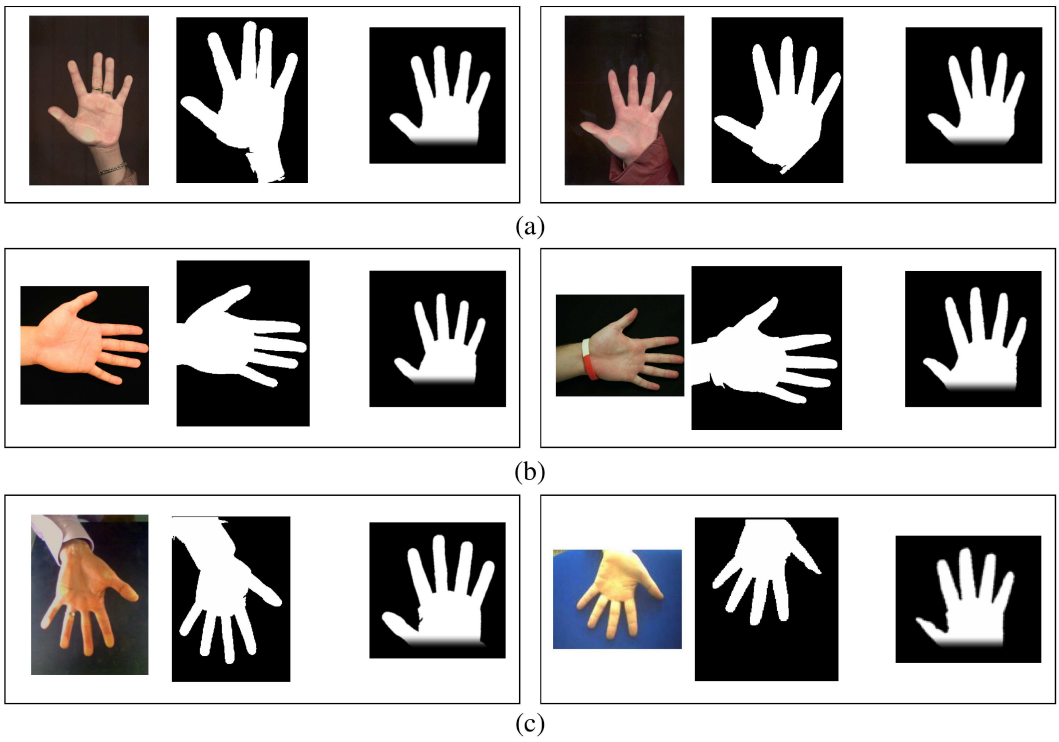


Figure 3.2. Results of our segmentation and normalization algorithm for the original hand images of six different persons acquired from two different scanners (a), two different cameras (b), and two different low-resolution webcams (c). First column: acquired image; second column: binarized hand; third column: normalized hand.

We designed a peg-free segmentation and normalization algorithm that operates with a large range of imaging devices, under varying illumination conditions and in the presence of hand accessories and sleeves. We impose only two requirements: (i) The background should be relatively homogeneous; (ii) Fingers should not touch each other. Figure 3.2 shows hand images acquired with scanners, cameras, and low resolution webcams, and the outcomes of our segmentation and normalization algorithm. The outcome quality of the segmentation and normalization algorithms is independent of imaging devices (scanner or camera) and of any special setup (special illumination, peg usage, etc.).

Figure 3.4 shows the block diagram of our novel hand-normalization scheme

along with the illustrative outputs of the intermediate steps. It involves the steps to segment the hand region via K-means clustering, morphological correction and ring or bandage artifact removal. Morphological operators mop up the holes in the foreground and debris in the background. The presence of rings or bandages on the finger is detected, and the silhouette is corrected with an ”artifact removal” algorithm [21, 34]. Finally, the hand and fingers are aligned to fixed orientations.

3.4.2. Hand Normalization

For hand biometry algorithms that utilize non-local features hand normalization is the most critical step. Hand normalization implies positioning of the global hand and orienting the fingers to fixed positions.

Jain and Duta [47] separately align pairs of corresponding fingers between the probe and gallery hand using a quasi-exhaustive polynomial search. Using the correspondences obtained from the finger alignment search, they apply Procrustes analysis and declare the mean alignment error as a measure of distance between two hands. Wong and Shi [41] implemented an alignment algorithm using nine landmarks (finger tips and valleys), which are in turn detected with the extrema of the hand contour curvature. The middle finger baseline is obtained by the straight line connecting the two valleys around the middle finger. The palm is rotated to a common reference frame according to an axis formed on the middle finger. Then the other fingers are rotated to align with those of a template hand, with matching middle fingers. Kumar et al. [44, 45] approximated the binarized shape of the hand by an ellipse. They used moments of the binary hand to extract the best-fitting ellipse. The hand is rotated according to the angle of the major axis of this ellipse. This aligned silhouette is then used for computing geometrical measures of the hand and for localizing the palmprint region.

The alignment of purely palmprint-based schemes is somewhat different. For example, Zhang and Shu [50] claimed that the three datum points are rotation invariant and can be used to construct a local coordinate system for alignment of

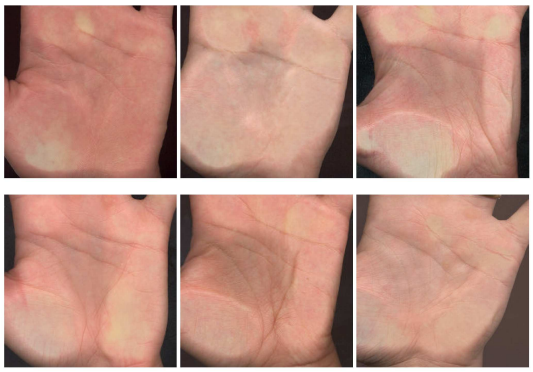


Figure 3.3. Palm images where datum points cannot be determined precisely.

line features. These references are the endpoints of the heart line and of head line while intersecting with the sides of the palm and their midpoint. Obviously this alignment algorithm is not robust since the assumptions that the life and head lines extend till one side of the palm and that life and head line merge before ending on the side of the palm do not hold for a non-negligible portion of the population (Figure 3.3). The authors report that this alignment scheme failed in five per cent of the images.

Zhang et al. [62, 63, 64, 65, 80] have proposed a more robust palm extraction and alignment algorithm based on the finger valleys. Once the two finger valleys, between the index and middle fingers and between the ring and pinky fingers, are detected, the line connecting these two crotches constitutes the y-axis of the palmprint coordinate system. The mid-point corresponds to the origin and the perpendicular line through the mid-point is used as the x-axis. The palm image's local coordinate system is rotated to align with a reference coordinate system and a central subimage is cropped as the aligned palm region of interest.

Our hand normalization algorithm [21, 34] minimizes intra-person variability of the hand postures, finger orientations and illumination, as illustrated in Figure 3.4. Briefly, the hand is translated and rotated to a reference frame, illumination correction is performed on it and the fingers are rotated around the pivot locations to preset orientations. The details of the normalization procedure can be found in

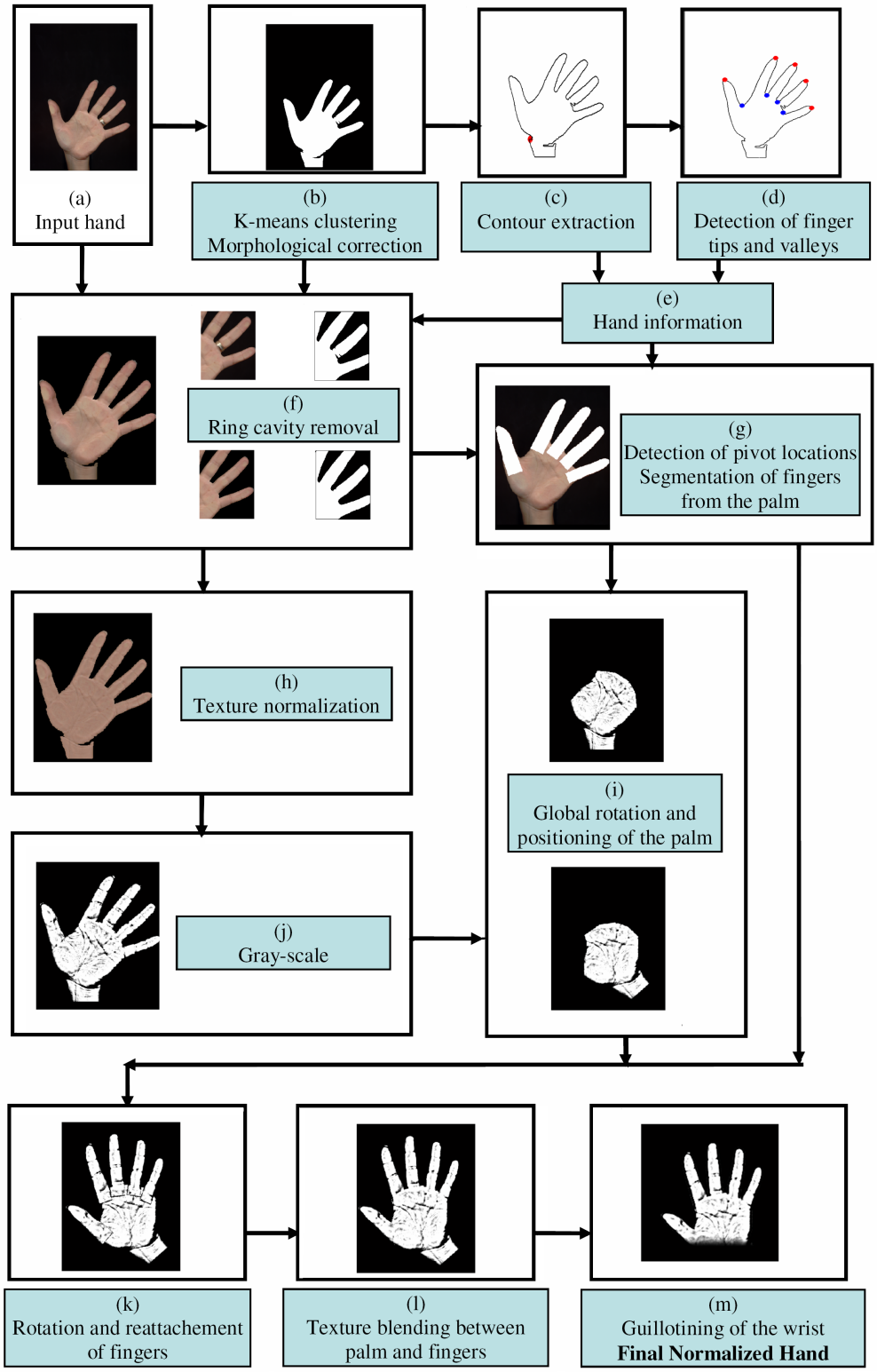


Figure 3.4. Block diagram of our hand normalization algorithm with illustrative intermediate outcome images.

our previous work [34]. The key points and the superiorities of the algorithm can be listed as follows:

- *Robustness to hand accessories*: Users are not obliged to remove their accessories, such as rings, clocks or bracelets, in this hand-based access system. The segmentation procedure of our normalization algorithm includes a ring and other artifact (like bandage) removal stage.
- *Texture correction*: Any non-uniform illumination effects and discolorations due to pressure applied by the user are corrected. First, the hand texture is converted to gray-level by choosing the principal component color with the largest variance. Second, the artifacts due to the non-uniform pressure are removed by a Gaussian-kernel high-pass filtering.
- *Finger rotation around pivots and texture blending*: We estimate the pivot locations (see Section 3.2.1), which are the joints between proximal phalanx and the corresponding metacarpal bone, corresponding to the knuckle positions on the reverse side of the hand. The pivots provide robust reference points around which the fingers can be rotated to pre-determined directions. The palm texture around these finger joints is corrected to avoid any artificial texture discontinuity due to rotation.
- *Palmprint extraction*: Palmprint extraction is a by-product of our normalization algorithm. A rectangular region inside the palm is extracted with the use of pivot locations. The details of this extraction procedure are given in Section 3.5.3.
- *Wrist guillotining*: The wrist region is guillotined at a certain latitude, which also removes any shadows, cuff artifacts, foreshortening due to non-flat parts of the wrist. The wrist region is tapered off with a cosinusoidal window that starts from the half distance point between the pivot line and the wrist line.

Our normalization algorithm can process hands acquired at very different conditions (Figure 3.2). The success of the algorithm is 100 per cent, in that all of the hands in our database were successfully normalized. The normalization procedure supplies the proper input format for subsequent feature extraction schemes, from geometrical

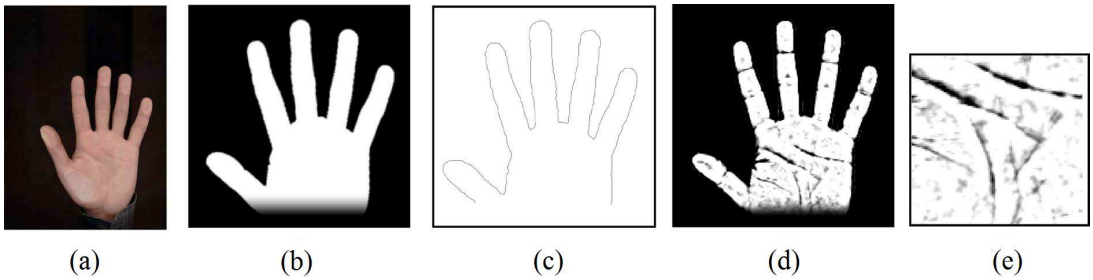


Figure 3.5. (a) Original hand, (b) Normalized binary hand, (c) Hand contour, (d) Global hand appearance (handprint), (e) Palmprint.

measures to statistical shape analysis tools, from subspace methods to palmprint-based feature extraction schemes.

The outcome of this algorithm is the normalized hand, which in turn, can be given as shape in binary form, as contour information, or as global hand appearance. The global appearance is referred to as the ”handprint”. The normalization procedure also includes the extraction of the palmprint region (Figure 3.5). In the next section, we briefly describe the features extracted from these ”modalities” of the normalized hand.

3.5. Hand and Palm Features

3.5.1. Geometrical Hand Features

Although the focus of our work is holistic hand features, we have also made tests with our own geometrical features for two reasons. First, our hand normalization algorithm provides by-product key information, such as locations of finger extremities and pivot positions that can be used to extract geometrical features. Second, the comparative performance of geometric features was not ever assessed on a database of this size (918 subjects), which is an order of magnitude larger with respect to other test databases in the literature. Our geometrical set consists of 28 features, some of which are illustrated in Figure 3.6:

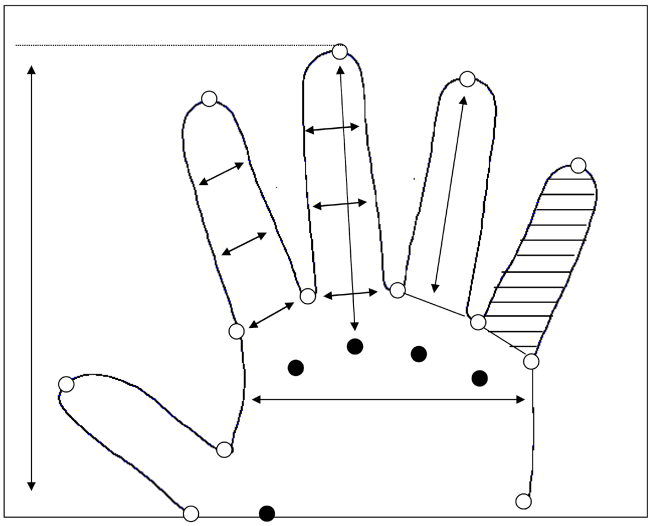


Figure 3.6. The geometric measures used for test on our database.

- Five finger lengths computed from the midpoints of the finger baselines to the finger tips. A finger baseline corresponds to the line connecting the two valleys around the corresponding finger;
- Fifteen finger widths measured, respectively for each finger, at the baselines, at one third of the length up, and at two third of the length of the fingers;
- Five finger areas;
- The palm width;
- The length of the hand;
- The total area of the hand.

3.5.2. Shape Features

We have considered several features that represent the global shape of the hand. These are extracted either from the binary hand or from the hand contour.

3.5.2.1. Pixel Difference of Binary Hands. The pixel difference of binary hands is the sum of the absolute difference of two binary hand images. This simple comparison technique provides a measure of the success of the hand normalization algorithm

in mitigating the shape variations due to hand posture and finger orientations. We intend to use it as a baseline against which the gain of the subspace methods can be measured.

3.5.2.2. PCA of Binary Hands. Each binary hand is organized in a single one-dimensional vector, and then the collection of vectors in the training database is subjected to principal component analysis. The PCA bases correspond to the eigenvectors of the covariance matrix of the hand vectors. The N-dimensional feature vector of a hand is obtained by projecting it onto the principal N eigenvectors.

3.5.2.3. ICA of Binary Hands. We apply the ICA analysis tool on binary hand images to extract and summarize prototypical shape information. ICA assumes that each observed hand image is a mixture of a set of N unknown source signals. We first apply PCA to the training set of binary images to reduce their dimension to N. Then we implement the ICA2 algorithm, which finds a linear transformation that minimizes the statistical dependence between the mixing coefficients.

3.5.2.4. ART of Binary Hands. We have defined the Angular Radial Transform (ART) in Section 2.4. We define the binary image in polar coordinates as $f(\theta, \phi)$, then obtain $N \times M$ ART magnitude coefficients $\{F_{mn}\}$ by projecting the image onto the ART basis functions up to order M and N. In shape recognition, the ART coefficients are normalized to F_{00} in order to achieve scale invariance; in our work we specifically make us of this coefficient for discriminatory size information. After aligning the hand images and placing them in a fixed-size image plane, we take the center of the plane as the center of the unit disk. Furthermore, each pixel location is converted to polar coordinates and the radial coordinate is normalized with the image size to have a value between zero and one.

3.5.2.5. Distance Between Contours. The contour of the normalized binary hand is another representation of the hand. Let us represent the hand contour vector

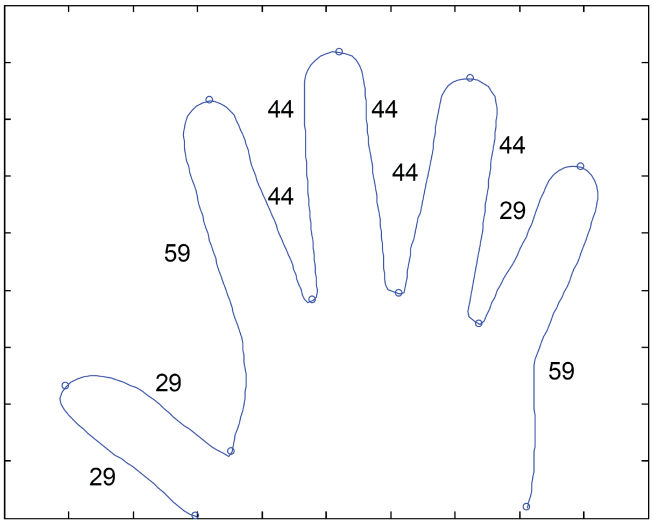


Figure 3.7. The number of points between landmark positions in the re-sampled hand contour.

of length $2n$ as $\mathbf{z} = (c_x(1), \cdots, c_x(n), c_y(1), \cdots, c_y(n))$ where n is the number of points of the hand contour and $(c_x(i), c_y(i))$ are the 2D coordinates of the ith point on the contour. We first establish the nine fiduciary reference points. We first establish 11 fiduciary reference points, consisting of the first and last contour elements, the five finger tips and the four finger valleys, and then resample the contour data in order to guarantee correspondence between contour elements of all hands. The number of samples between two landmark points is kept equal for all hands; hence the sampling step sizes differ proportionally to the hand size and shape. Figure 3.7 gives the number of contour elements chosen between landmarks of the hand. Notice that we exclude from the contour the horizontal line above the wrist. In total, hand contours have 435 points. The difference between two hand contours is the sum of the absolute difference between the coordinates of the corresponding points.

3.5.2.6. PCA of the Contours (Active Shape Modeling). The covariance matrix $\mathbf{C}$ of the contour vectors is constructed as:

$$\mathbf{C} = \frac{1}{s-1} \sum_{i=1}^{s} (\mathbf{z}_i - \hat{\mathbf{z}})(\mathbf{z}_i - \hat{\mathbf{z}})^T \tag{3.1}$$

using the s sample hands, and where $\hat{\mathbf{z}}$ is the mean contour vector. The eigenvectors $\{\mathbf{u}_i\}$ of the covariance matrix sorted in decreasing order with respect to the corresponding eigenvalues $\{\lambda_i\}$ model the variations in the training set. If $\mathbf{U}$ contains the K eigenvectors corresponding to the largest eigenvalues, then any shape vector in the training set can be approximated as $\mathbf{z} \approx \hat{\mathbf{z}} + \mathbf{U}\mathbf{b}$, where $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_K]$ is the selected eigenspace basis set and $\mathbf{b}$ is the projection of shape $\mathbf{z}$ to this eigenspace, i.e. $\mathbf{b} = \mathbf{U}^T(\mathbf{z} - \hat{\mathbf{z}})$. The vector b serves as the feature vector of length K of a hand contour in the matching stage.

Figure 3.8 shows the effect of varying the first 10 modes of $\mathbf{b}$, one at a time. The shapes in this figure are obtained by summing a perturbed nth eigenvector with the mean shape vector. The perturbations are exaggerated intentionally to make the effect of the corresponding mode more visible. A comment is added below each figure inset related to the major visible effect of eigenvalue perturbation, though especially for higher eigenvalues, multiple effects can occur.

3.5.2.7. DFT of the Contours. Fourier descriptors are efficient features for shape characterization due to their scale, translation and rotation invariance, as well as due to their immunity from small shape perturbations. Fourier descriptors are derived from the Discrete Fourier Transform (DFT) coefficients of a closed contour that is represented as a periodic complex function. We represent the hand contour as a complex function, where x-coordinates form the real part and the y-coordinates the imaginary part. We use the first K DFT coefficients as features, and K varies between 15 and 50, depending on the number of classes (subjects). We do not apply any normalization on the coefficients, since our hand contours are already pose-normalized. The real and imaginary parts of the raw DFT coefficients are concatenated to form a feature vector of size $2K - 1$.

3.5.2.8. Distance Transform Features. In the shape-based retrieval of objects based on their 2D views, as proposed by Funkhouser et al. [28] first, the distance transform (DT) on the planar shape is calculated, and this is followed by sampling of the DT

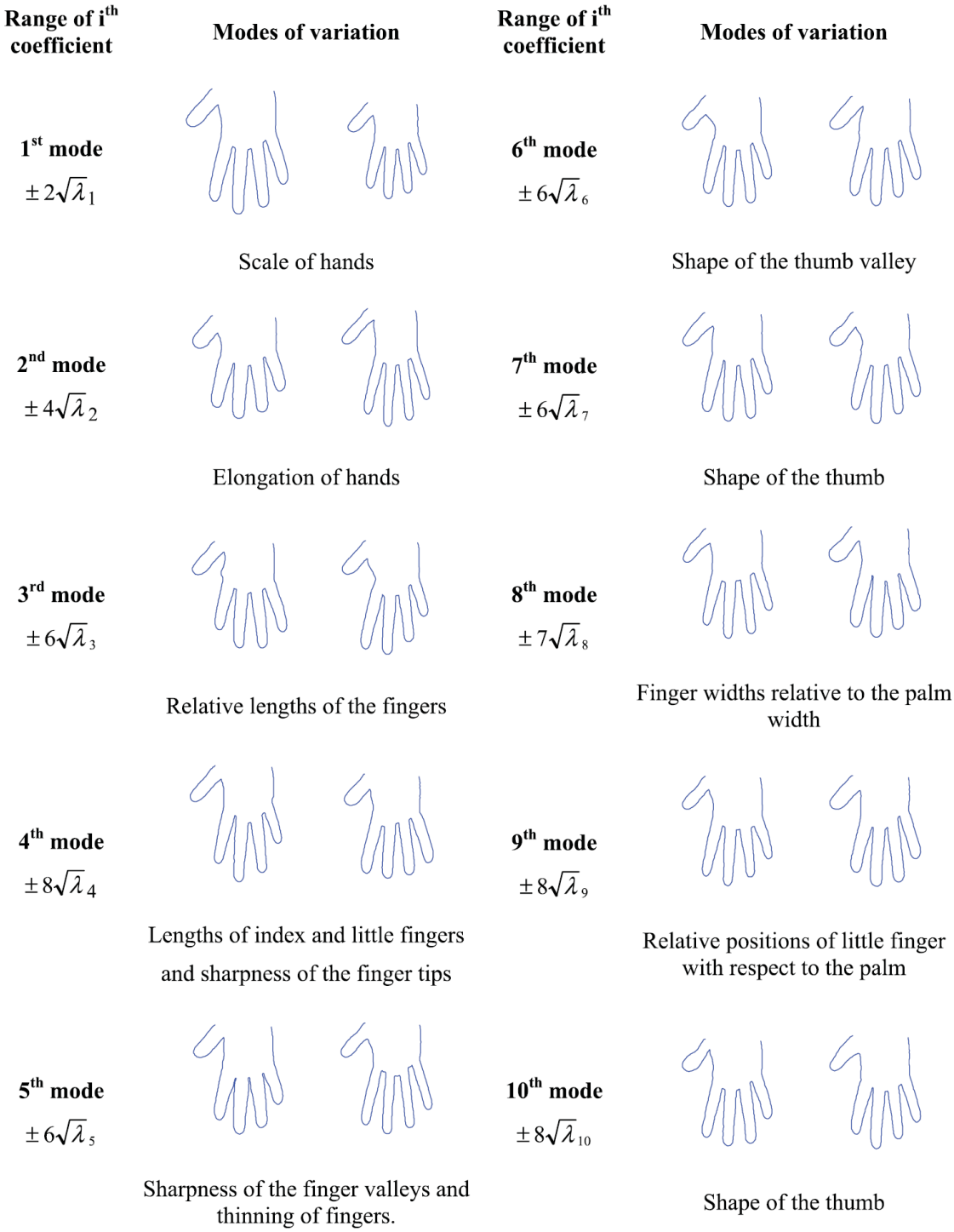


Figure 3.8. Effect of varying the weights of the first ten eigenvectors.

surface with concentric circles (Figure 3.9). The 1D periodic mass (say, one for hand region, zero for background) on the circles is subjected to the DFT and a shape signature is obtained by considering a selected number of low-order DFT magnitude coefficients. Thus, these features are indexed both by the circle number and DFT coefficient number. As in the case of ART features, the center of the circles is positioned on the center of the plane. The span of radii is constant for all hands. This feature applies obviously only to the shape information, and not to the texture.

Figure 3.9 a and b show the contour of a hand image and its distance transform. Figure 3.9 c and d show, respectively, the concentric circles drawn and the resulting profiles. Finally, Figure 3.9 e illustrates the feature extraction scheme.

3.5.3. Palmprint Features

We utilize the pivot locations extracted in the hand normalization step to localize and scale the palmprint region (Figure 3.10). The line connecting the pivots of the index and little fingers constitute the upper side of the rectangle. The rectangle is extended until it intersects the parallel line passing through the pivot location of the thumb. The region is than resized to a fixed size image with linear interpolation.

We extract PCA and ICA-based features from the palmprint image. The PCA-based approach is known as eigenpalm approach and implemented by several authors [67, 68, 69, 70, 87]. The PCA and ICA-based feature extraction procedures are as described in Section 3.5.2; the only difference is that we form the data vectors from the palm images.

3.5.4. Global Hand Appearance

In order to incorporate the texture and shape information of the hands, many authors have proposed fusion methods at feature and score levels [45]. These schemes involve separate treatment of each modality, i.e. the shape and palm features are extracted separately and are in general of different nature. For example

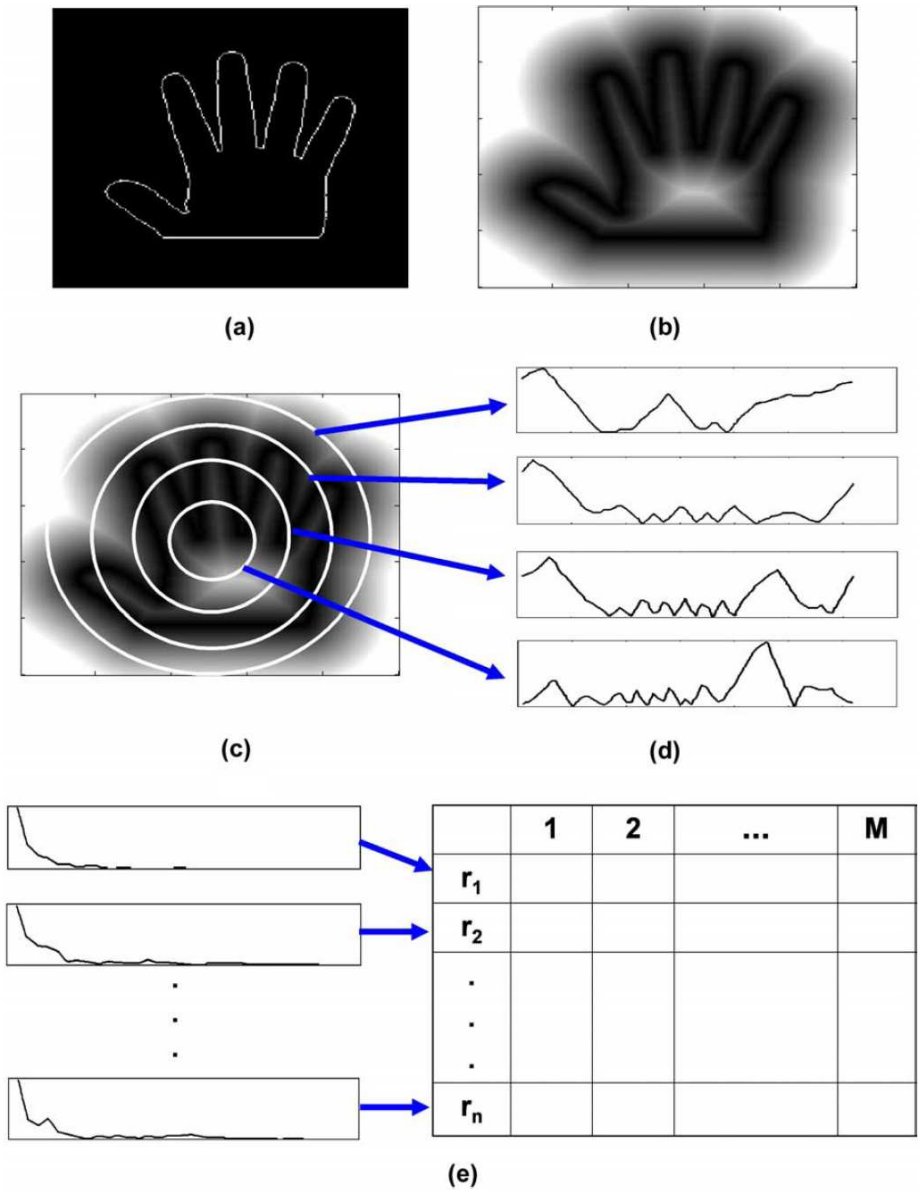


Figure 3.9. (a and b) Contour of a hand and its distance transform defined on the plane. (c and d) Concentric spheres on the distance transform and extracted profiles on circles. (e) Feature extraction: DFTs of the circular profile of the distance transform function and the selected coefficients.

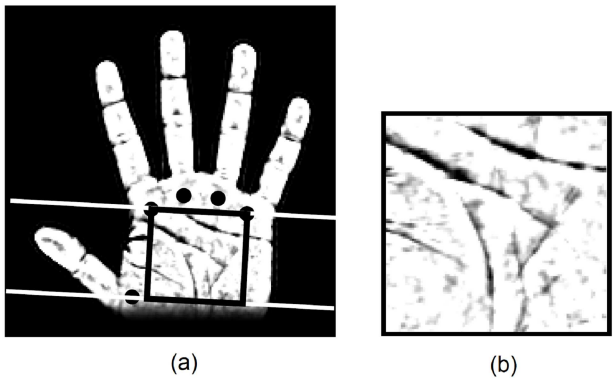


Figure 3.10. Extraction of the palmprint region. (a) The rectangular region determined by pivot locations, (b) The extracted palmprint.

Kumar et al. [45] have fused geometric features representing the shape and the Fourier coefficients extracted from the palm.

In our study, we make use of the "handprint", the outcome of our normalization algorithm, in order to extract features that inherently represent shape and texture jointly. The handprint contains the palm texture, finger creases and the silhouette of the normalized hand.

Recall that the hand normalization stage outputs a binary hand image, I_{shape}, as well as gray-scale textured hand image, $I_{appearance}$. The gray-level values of the hand texture are normalized to have unit mean and unit variance. Then, either the binary shape image or its textured version is fed to the ICA feature extractor, as illustrated in Figure 3.11. The composition of shape and texture components can be adjusted by altering the weighting factor, or texture to shape ratio, denoted as α :

$$I = I_{shape} + \alpha I_{appearance}, \quad 0 \leq \alpha \leq 1 \tag{3.2}$$

By reducing the weighting factor, the contribution of the texture component is attenuated. In fact, when it is set to zero, the input to the feature extractor becomes pure shape; i.e. the normalized hand silhouette.

Figure 3.11. Weighted combination of shape and texture components.

3.5.5. Active Appearance Modeling

We have followed Cootes method [44] to decouple texture information from shape. To this end, each image is warped to make its landmarks match with those of some mean shape. Thin-plate splines are used for image warping as in Bookstein [45]. The resulting warped texture information is then expressed as a 1D vector. Finally, PCA is applied to the texture vectors of the training hand examples to obtain modes of variation of the texture.

Let $\mathbf{b}_h$ be the projection of a hand to the shape eigenspace and $\mathbf{b}_g$ the projection of the warped hand to the texture eigenspace. The vector $\mathbf{b} = [\mathbf{b}_h \ \mathbf{b}_g]^T$ serves as the feature vector of the hand. The dimensions of both shape and texture eigenspaces are important parameters and are optimized through experimental work. The distance between two hands are computed using a weighted sum of squared differences of feature vector components. When matching is performed using only shape information the distance between two feature vectors, $\mathbf{b}^k$ and $\mathbf{b}^l$, is:

$$D(k,l) = \sum_{i=1}^{K} \frac{1}{\sqrt{\lambda_i}} \left(\mathbf{b}_i^k - \mathbf{b}_i^l\right)^2 \tag{3.3}$$

When matching is performed using shape and texture information together, the distance is:

$$D(k,l) = \sum_{i=1}^{K} \frac{1}{\sqrt{\lambda_{hi}}} \left(\mathbf{b}_{hi}^k - \mathbf{b}_{hi}^l\right)^2 + \sum_{i=1}^{L} \frac{1}{\sqrt{\lambda_{gi}}} \left(\mathbf{b}_{gi}^k - \mathbf{b}_{gi}^l\right)^2 \tag{3.4}$$

where $\{\mathbf{b}_{hi}^k\}_{i=1}^K$ are the K-dimensional shape features of the kth hand, $\{\mathbf{b}_{gi}^k\}_{i=1}^L$ are the L-dimensional texture features of the kth hand, and λ_{hi} and λ_{gi} are the ith eigenvalues obtained from PCA of shape and texture vectors, respectively. The squared difference of each feature is divided by the square root of the feature variance as observed in the training set.

3.5.6. ART of Hand Appearance

We also compute the ART coefficients for the shape plus texture appearance data, which includes palm and finger gray-level details. The computation of the ART coefficients is similar to that with silhouette hand images.

3.6. Experimental Results

In this section we report our novel performance results of hand biometry, with and without texture. We give performance figures with respect to various hand features. We address the relative contributions of shape and texture, fusion schemes of right and left hands at various levels, the generalization ability of the ICA-based scheme, the time lapse issue and robustness to the resolution of hand images.

3.6.1. Hand Database

Our database contains hands from 918 subjects acquired with flatbed scanners within four years. No positioning aids were used. The users laid their hands comfortably on the scanner in any orientation with the only constraint that their fingers are kept apart. Users were not required to take off their accessories such as rings and watches. All the images were originally scanned at 150 dpi and reduced to 45 dpi via bilinear resizing. None of the users or their images was discarded.

Table 3.1 gives a summary of the properties of the database. The database is organized in five sets. Set A contains left hands of 918 subjects while set B contains ambidextrous recordings, that is, 800 subjects out of total of 918 have both left and

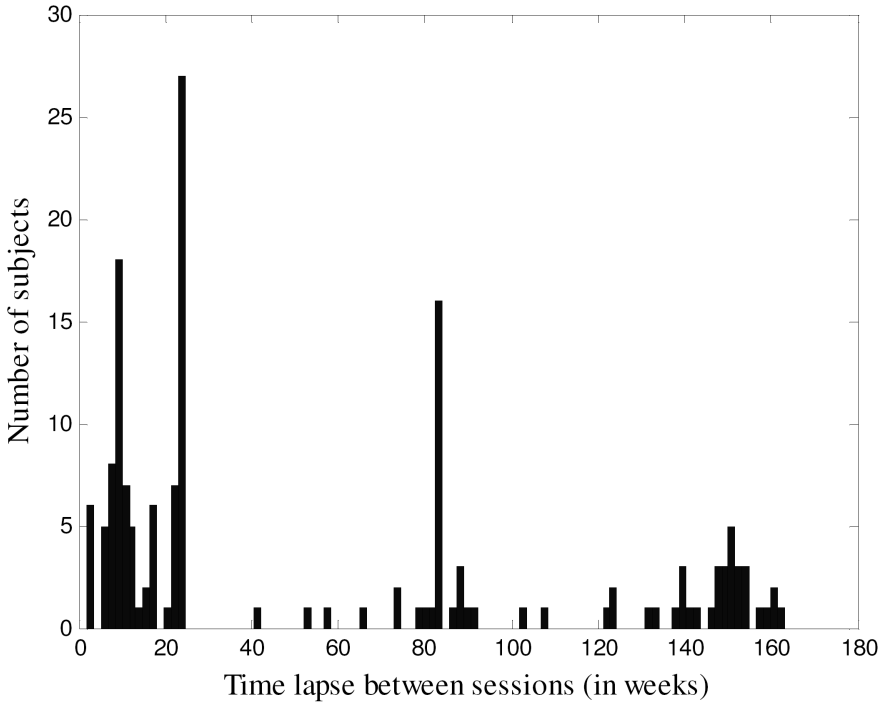


Figure 3.12. Histogram of the time lapse between two sessions of the subjects in hand data set C.

Figure 3.13. Hand images of six subjects. First row contains first session hands. Second row contains hand images of the same six subjects acquired after time lapse varying between two weeks and three years.

right hands images. The subjects in set C and D form a subset of those A and B, whose hand images were re-acquired after a time lapse varying from two weeks to three years. Figure 3.12 shows the histogram of the time lapse between two scanning sessions of the subjects in set C. The average time lapse is one year. In set C, only left hands are present, whereas set D contains time-lapse re-scanned left and right hands of 100 subjects. The effect of time lapse is demonstrated in Figure 3.13 where the second row contains the later hand images of the six subjects in the first row. Finally, in set E, there are left hands of 458 subjects. This set is a subset of Set A.

Table 3.1. The properties of the hand database.

Set	Hand type	# subjects	# samples/subject	Time lapse
A	Left	918	3	Short
B	Left and Right	800	2x3	Short
C	Left	160	3+2	One month to two years
D	Left and Right	100	2x3+2x3	One month to two years
E	Left	458	3	Short

3.6.2. Performance and Feature Types - Part I

In this section we compare the performance of the feature types listed in Table 3.2, which gives the rank-one identification performance with these features under changing population size. For each population size, random subsets were drawn from the largest set, i.e. from set A, and the gallery and test images were interchanged leading to multiple experiments. The average performance of these experiments is reported in Table 3.2. We considered four different representation types, namely: (i) Hand contours; (ii) Shape of the hand silhouette, called also binary hand; (iii) Palmprint image extracted from a rectangular window on the palm; (iv) Hand appearance, the hand texture bounded by the hand silhouette shape. A number of conclusions can be drawn from these figures:

- *Raw data versus PCA subspace data*: We see that PCA, when applied to the hand appearance data, brings negligible performance advantage, and for large

Table 3.2. Identification performances with respect to the feature type and the population size. Enrollment size is two; only left hands are used.

Population size:		**50**	**100**	**200**	**400**	**600**	**918**
Number of experiments:		**180**	**90**	**60**	**30**	**15**	**3**
Shape	Geometric measures + LDA	98.36	98.77	98.22	97.79	97.71	97.49
	Point set difference of the contours	98.28	97.49	96.24	94.56	93.83	92.88
	Pixel difference	98.39	97.90	96.97	96.77	95.53	95.03
	PCA on binary hands	98.44	98.00	97.28	96.10	95.61	95.21
	ICA on binary hands	99.49	99.34	98.99	98.21	98.71	98.69
	DFT on contour + LDA	98.41	99.34	99.44	99.38	99.23	99.31
Palm texture	PCA on palm texture	95.31	94.73	93.76	92.82	92.50	91.98
	ICA on palm texture	95.59	95.10	93.88	91.79	93.31	93.83
Appearance	Pixel difference	99.34	99.29	98.89	98.33	98.23	97.93
	PCA on appearance	99.06	98.73	98.18	97.46	97.19	96.66
	ICA on appearance	**99.73**	**99.74**	**99.52**	**99.40**	**99.44**	**99.42**

populations causes even some small performance loss. Its only advantage is in reducing hand image data by approximately two orders of magnitude. In other words, from the image size (200x200) down to the population size, since we can at most get that many independent columns. It is also noteworthy that ICA always outperforms PCA by two to three percentage points. This is in contrast to the face literature where ICA and PCA were reported to have similar performance [91].

- *The top performing feature*: The top-performing feature was found to be ICA (Architecture II) operating on the hand appearance data. This is closely followed by ICA-II features operating on binary shape and DFT coefficients of hand contour data with linear discriminant analysis. The addition of texture information to ICA-scheme (binary versus textured hands) proves especially beneficial for large population sizes.
- *Discriminant analysis*: We have applied LDA (Linear Discriminant Analysis) on geometric measures and on DFT coefficients of the contour and these are the only feature types that benefited from the class information in the enrollment

phase. The reason to use LDA for geometric features was the fact that they were very disparate in size (areas, lengths etc.) and LDA contributed to their normalization. With this advantage, geometric measures give fairly good results as compared to other shape-based methods, with the exceptions of the DFT and ICA features.

- *Shape contour versus shape alpha-plane*: We have observed that point set difference of the contours yielded relatively poor identification results. The first reason is that small variations in hand shape have more impact on contour information than on the binary image. Second, the hand contour samples are not in perfect correspondence, since we just apply uniform sampling between the eleven landmarks (five finger tips, four finger valleys, first and last points of the contours). In contrast, the binary hand silhouette (shape alpha plane) yielded consistently better results.
- *DFT coefficients*: DFT coefficients of the contours give good identification performance which is very close to that of the ICA-based method. The main reason of this high performance is that we apply LDA on the raw DFT coefficients, and LDA re-weights these coefficients such that maximum class separation is obtained for the training samples. Furthermore, the Fourier descriptors smooth out the small shape variations on the contour irrelevant to class characteristics and ignore correspondence mismatches among different hands. The high performance yielded by the DFT coefficients show the success of our hand normalization algorithm and strengthens our claim that the shape of the hand contour contains richer information than the geometric measures.
- *Palmprint-based features give the worst results*: We have observed that the varying amount of stretching in the palm from session to session and the contact flattening causes folds on the mass of the palm, and displaces the palm lines resulting in misalignment between palm features. Our performance figures are comparable to the state-of-the-art palmprint recognition from low resolution images. For example, Kumar and Zhang [46] reported 95.8 per cent classification rate of palmprints with a population of size 100. We have obtained 95.1 per cent recognition rate on average with 90 different sets consisting of 100 subjects. With increasing populations, the discriminating ability of palmprint

features reduces to unacceptable levels. This means that, unless palmprint data are collected with specialized equipment as developed by Zhang et al. [62], the data will have mediocre reliability.

3.6.3. Performance and Feature Types - Part II

In this section we compare the performance of the feature types listed in Table 3.3, which gives the rank-one identification performance with these features under changing population size. The experimental setup is similar to Part I. The difference is that we have used a smaller data set (set E) to evaluate the identification performances.

As in Table 3.3, the population size grows an order of magnitude from 40 to 458 all features suffer a performance drop ranging from one to three per cent. The only exception is the ICA features on appearance data, where the performance drop is only a meager 0.2 per cent, which again points out to the robustness of the ICA features.

Since we have established that the Independent Component Analysis features yield superior performance compared to all others, we have conducted the following experiments with different setups solely with ICA features.

3.6.4. Contribution of Shape and Texture

We can control the contribution of texture relative to the hand silhouette by adjusting a weighting parameter, as explained in Section 3.5.4. This weighting parameter is the ratio of the gray-level variation of the handprint to the level of the binary hand shape. Figure 3.14 gives the identification performance with ICA features for varying texture-to-shape ratio α. The database is set A, which contains left hands of 918 people. ICA-based features are extracted for classification. When we use only binary silhouette the performance is 98.69 per cent. As we increase the texture-to-shape ratio from 0 to 0.3, the performance increases and reaches its

Table 3.3. Identification performances with respect to the feature type and the population size. Enrollment size is two; only left hands are used. Set E is used.

Population size	40	100	200	458
Number of experiments	30	12	6	1
ICA on binary hands	99.19	99.09	98.55	98.40
ICA on appearance	99.68	99.65	99.58	99.49
PCA on contour (Active Shape Modeling)	98.67	98.69	98.56	97.19
PCA on contour and texture (Active Appearance Modeling)	99.14	98.89	98.72	97.99
ART on binary hands	98.72	97.78	97.00	95.78
ART on appearance	99.28	98.72	98.06	97.67
DT on contour	99.17	98.22	96.22	95.99

maximum value of 99.42 per cent. We encounter a broad maximum; and increasing the texture component beyond $\alpha = 0.9$ degrades the performance slightly down to 99.27 per cent.

3.6.5. Fusion of the Left and Right Hands

If both right and left hands are measured, several fusion opportunities arise. First, with our precise registration algorithm we can fuse the right and left hands at data level through averaging them. Notice that a right hand is simply flipped over horizontally, normalized and summed with its corresponding left hand. The second alternative is fusion at feature level, where two different ICA-vectors are constructed for right and left hands, and then these feature vectors are concatenated. The third alternative is to use fusion at score level. We have implemented and compared score level fusion with max and sum rules.

We have conducted experiments on the database of size 800 (set B) using the ICA-based features extracted from global hand appearance. Table 3.4 gives the identification performances of the single hand versus both hands fused in various styles of data, feature and score. The main observations are:

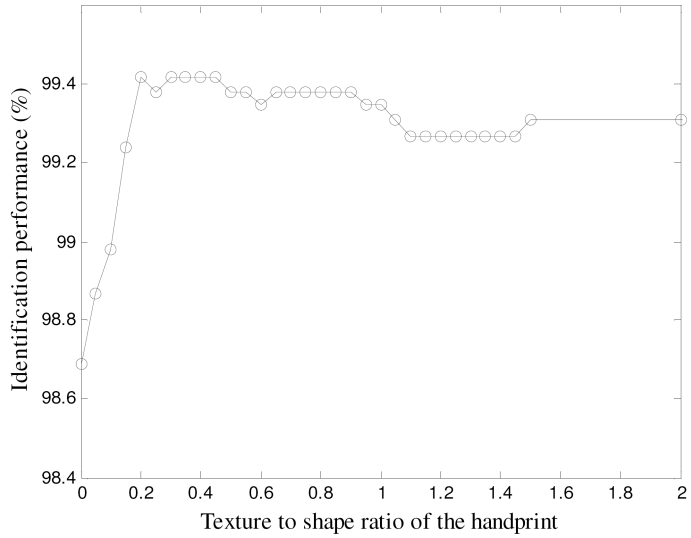


Figure 3.14. Identification performance as a function of texture-to-shape ratio α. The population is of size 918 (Set A).

- When we average normalized gray-level appearances of the right and left hands, the performance improves by two points from 97.74 to 99.63 per cent for single enrollment, and by 0.60 points from 99.28 to 99.88 per cent for double enrollment. Notice that in double enrollment we take the average of four hands.
- However, to be fair we have to compare equal amounts of data. Thus when we compare "single hand and double enrollment" situation with "double hand and single enrollment", the advantage of ambidextrous biometry is much less impressive. The performance differential becomes 0.35 points. In other words, we can avoid the discomfort of ambidextrous access control simply with multiple enrollments.
- Finally, if subjects have two training samples per hand and get enrolled ambidextrously, the performance climbs to 99.92 per cent. This means that only one person in 800 is not recognized. These experiments were conducted in three folds by interchanging the gallery and probe hand images; and one out of the three experiments ended up with 100 per cent recognition rate, and in the other two experiments only one hand was misclassified. The misclassified

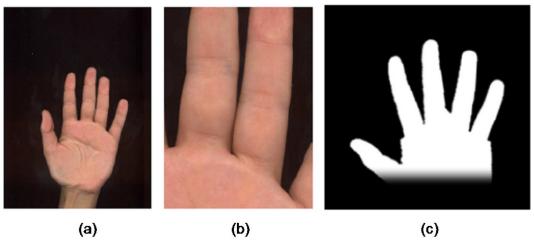


Figure 3.15. The misclassified hand (a), its zoomed version (b), and its normalized shape (c).

hand and its normalized version are shown in Figure 3.15a and Figure 3.15c respectively. Obviously this is a faulty image where the two fingers are not sufficiently kept apart as shown in Figure 3.15b.

- Score fusion under sum rule seems to perform slightly better than score fusion under max rule or data fusion. Note that for score fusion, left and right hands are considered separately, each having its own subspace.
- Feature fusion also gives slightly better results than data fusion. Feature fusion necessitates separate subspace building phases for left and right hands, and each hand is separately projected to either the left or right subspace. Then the projections are concatenated and a feature vector of double size is obtained. Thus, feature fusion is computationally more expensive than data fusion.

Despite the improvement of 0.35-0.50 percentage points on a population of size 800 in recognition performance, the employment of both right and left hands in a practice is disputable due to the increased user discomfort [90]. Finally, it is conceivable to have a system that accepts both right and left hands. The system must enroll subjects ambidextrously, and will operate on the left-hand or right-hand mode according to the placement of the test hand in the device. This choice would be a convenience for right-handed and left-handed people, for people with occasionally injured and bandaged hands, or simply when one of the hands is busy holding other objects.

Table 3.4. Identification performances with left and right hands and with the fusion of right and left hands. The population is of size 800 (Set B). ICA-based features of global hand appearance are used.

	No fusion		Fusion of left and right hands			
Enrollment Size	Left	Right	Data fusion	Feature fusion	Score Fusion (Max rule)	Score Fusion (Sum rule)
1	98.00	97.48	99.63	99.65	99.40	99.73
2	99.42	99.13	99.88	99.92	99.92	99.92

3.6.6. Generalization Ability of the System

The generalization ability of a subspace-based method is defined as its capability to function with new data, that is, to serve as basis vectors for new data that were not used in the first place to construct the basis set, and it is important for three reasons: First, the subspace-building phase requires memory and computation time; hence it is undesirable to re-train the system every time a new user is registered to the system. Second, the system should be able to model unseen subjects, especially for verification tasks. Third, the subspace trained in one population should be exploitable for another population. Thus, the ICA basis vectors from one population of subjects should function as the basis set, providing a ready-to-use system for a new application without the necessity of collecting images to build a subspace.

We can classify the subjects into three sets: The training set, the gallery set and the impostor set. The training set contains images of the subjects that are used to build the subpace, in our case, the ICA-subspace. The gallery set consists of subjects that are registered to the system and are expected to be identified or verified. These two sets can be identical, totally different or intersecting. The impostor set is disjoint from the training and gallery sets and consists of unauthorized users that should be rejected by a verification system. We have conducted three different experiments in order to test the generalization ability of our ICA-based recognition system.

Table 3.5. Identification performances with respect to the size of the training set for building the ICA subspace. The gallery set is of size 918 and contains both seen and unseen subjects during the subspace-building phase.

	Number of Features						
	50	**100**	**200**	**300**	**400**	**500**	**600**
Training set size							
50	95.51						
100	95.85	97.69					
200	96.30	98.16	98.83				
300	96.81	98.39	98.99	99.14			
400	96.70	98.42	98.97	99.16	99.23		
500	96.67	98.65	99.07	99.24	99.27	99.32	
600	96.84	98.69	99.09	99.27	99.31	99.24	99.20
700	96.70	98.73	99.20	99.38	99.31	99.38	99.38
800	96.55	98.69	99.16	99.38	99.42	99.38	99.46
918	96.84	98.69	99.24	99.38	99.42	99.42	99.46

3.6.6.1. The Effect of Training Set Size. In the first experiment, the identification performances are calculated on a test set of 918 people (Set A), using various ICA-subspaces built with training sets of different sizes, each corresponding to a different subset of the set A. Hence the gallery set contains both seen and unseen subjects during construction of the ICA-subspace. For example, we use a randomly chosen subset of 200 hands to build the ICA-subspace and recognize persons in a set of 918 persons, without the contributions of the 718 remaining subjects for building the ICA basis vectors. Table 3.5 gives the results of the identification performance under various training set sizes and number of features. Five random combinations of training samples are drawn from the population and the identification experiment is repeated five times for training set sizes of 50 to 500 and the average identification performance is reported. For larger training set sizes, i.e. of 600 to 918, the experiment is carried out for only one combination of training and test samples. The number of features is chosen equal or less than the training set size since the dimensionality of the ICA-subspace is limited by the number of available images. We can make two observations: (i) For a fixed number of features, the performance

Table 3.6. Identification performances with respect to the size of the training set for building the ICA subspace. The gallery subjects are chosen from a population apart from the training subjects.

Training set size	Gallery set size	Identification performance	# misclassified images
50	400	96.35	15
100	400	98.45	6
200	400	98.95	4
300	400	99.28	3
400	400	99.42	2
500	400	99.40	2

deterioration with increasing training set size is marginal; (ii) The optimal feature size seems to be 300, as there is not much of an improvement for population sizes from 300 up to 918. For example, when we grow the training set size from 300 subjects to the maximum possible size, i.e. 918, and the feature components from 300 to 600, the number of misclassified samples only drops from eight to five.

3.6.6.2. Disjoint Training and Gallery Sets. In the second experiment the training and the gallery sets are totally disjoint. This is the case when the system is trained on a given population and then exported to another platform where totally different subjects use the system. The gallery set consists of 400 subjects. The identification performance increases incrementally after training set size of 100 (Table 3.6). For all training set sizes, we have drawn five random combinations of training and test samples and averaged the identification performances obtained from the five experiments. The results in Table 3.6 indicate that this biometric system is completely generalizable, in view of the uncompromising high identification performance.

3.6.6.3. Verification and Impostor Rejection. In the third experiment, we simulate a verification scenario, where the gallery set and impostor set consist of 400 and 100 subjects, respectively. We have 400 genuine-to-genuine and 100x400 impostor-to-genuine comparisons. None of the gallery and impostor subjects have been seen at

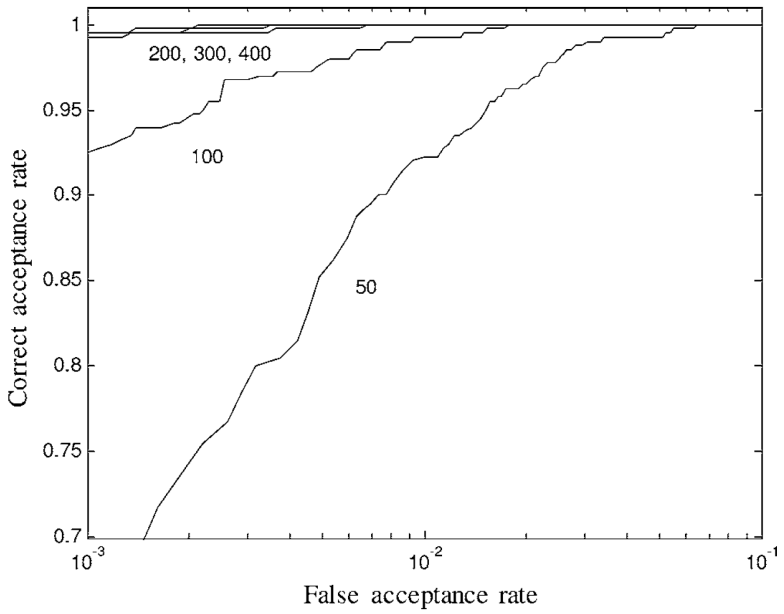


Figure 3.16. ROC curves with respect to the size of the training set for building the ICA subspace.

the subspace-building phase. The system is trained with different sizes of populations, as in the second experiment above. Table 3.7 gives the equal error rates. These error rates are averaged over five-fold experiments where combinations of training, genuine and impostor samples are selected randomly. We observe that after training set size of 100, the improvement is not significant. Figure 3.16 shows the receiver operating characteristics of the system; the ROC (receiver operating characteristics) curves of systems trained with 200, 300 and 400 subjects are hardly differentiable. We can conclude that the system has good impostor rejection performance.

These three experiments demonstrate that our ICA-based hand recognition scheme can model adequately hands that were unseen during the model-building phase. The trained subsets can be imported to other populations with identification rates higher than 99 per cent, and equal error rates lower than 0.4 per cent.

Table 3.7. Verification performances with respect to the size of the training set for building the ICA subspace. The gallery and impostor subjects are chosen from a population apart from the training subjects.

Training set size	Gallery set size	Impostor set size	EER (%)
50	400	100	1.24
100	400	100	0.66
200	400	100	0.40
300	400	100	0.27
400	400	100	0.21

3.6.7. Effect of Resolution on the Performance

We have tested our normalization algorithm and ICA-based feature extraction scheme under various image resolutions. All other experiments in this work were performed with 45-dpi resolution, and the resulting normalized images were of size 200x200. We reduced the resolution to 30 and 15 dpi via linear interpolation and conducted identification experiments on the set A (population 918). The rates of success for normalization and identification are separately given in Table 3.8.

When some images are downsampled to a lower resolution, fingers that are close to each other tend to merge, which makes the hand normalization impossible. There were two such hand images with resolution 30 dpi and six images with resolution 15 dpi, and they were discarded from the identification experiments. A sample case is illustrated in Figure 3.17.

This analysis shows that our normalization algorithm can work with very low-resolution images and the identification performance remains above 96 per cent even at 15 dpi.

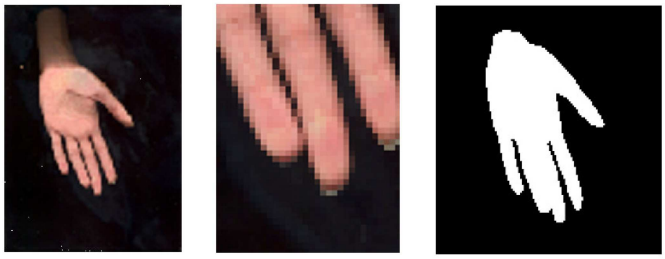


Figure 3.17. (a) Sample hand image at 15 dpi. (b) Zoomed hand image (c) Result of segmentation.

Table 3.8. Identification performances with respect to resolution. The population is of size 918 (Set A). ICA-based features of global hand appearance are used.

	45 dpi	30 dpi	15 dpi
Success of normalization (%)	100	99.79	99.35
Identification performance (%)	99.42	99.02	96.24

3.6.8. Performance under Time Lapse

Robustness with respect to time lapse is the most critical issue of a biometry-based identification system. Table 3.9 gives the identification rates obtained on a test set of 160 subjects (Set C). Hence we conducted experiments with time-lapse images, which were acquired after a period ranging from two weeks to three years. In the experiments, we varied the population size of the training set for building the ICA subspace and only "old images" were used. When we use only the old images of 160 subjects for training, we end up with four misclassified cases within recent test hands of these subjects. As the number of training images increase, the dimensionality of the subspace, hence the number of features increase, we achieve 100 per cent recognition rate. The last experiment in Table 3.9 corresponds to the case where new images of 160 people are compared with the full gallery of size 918 subjects, i.e. 918 classes exist. Even in this difficult setup, the identification performance is 99.06 per cent.

Table 3.9. Identification performances with respect to time lapse. ICA-based features of global hand appearance are used.

Training set size	**# users in the gallery**	**# test subjects**	**Identification performance %**	**# misclassified images**
160	160	160	98.75	4
300	160	160	99.38	2
918	160	160	100	0
918	918	160	99.06	3

Since there does not exist standard hand databases and protocols, it is difficult to evaluate the relative success of alternate works on different databases. However, in Table 3.10, we give the identification and verification results reported by several authors. In this table, we also indicate the key parameters of each experiment. The identification and verification performances are denoted as IP and VP, respectively. All the experiments are performed on databases consisting of 100 subjects since this was the population size common to the other studies in the literature in Table 3.10. We have conducted our experiments on set D, with the ICA-based features extracted from the global hand appearance. The number of test images is three for each subject. The verification results are obtained using an impostor set of size 100 subjects with three hand images for each, leading to 300 genuine-to-genuine comparisons and 300x100 impostor-to-genuine comparisons. Although the performance figures in Table 3.10 were obtained with different hand databases, we believe that they nevertheless give an idea of the success of the hand appearance based algorithm.

3.7. Conclusions

Our detailed investigation of the various aspects of hand biometry reveals that person identification and verification can be successfully implemented with hand imaging devices. Our major conclusions on the device technology and subject set list as follows:

- Proper hand registration with finger reorientations is critical for high perfor-

Table 3.10. Comparison of our method with previous work.

	Enrol. size	**Time Lapse**	**Hand type**	**Performance**
Kumar et al., 2006 [45]	5	Three months	Left	VP: 3.74 % FAR, 1.91 % FRR
Kumar and Zhang, 2006 [46]	5	Three months	Left	IP: 98 %
Kumar and Zhang, 2005 [67]	5	Three months	Right	VP: 0.08 % FAR, 4.6 % FRR
Shang and Huang, 2006 [76]	3	Two months	Right	IP: 98.67 %
ICA2 on handprint	3	Two weeks to three years	Left	IP: 99.33 % VP: 1 % EER
ICA2 on handprint	3	Two weeks to three years	Right	IP: 98 % VP: 1.16 % EER
ICA2 on handprint (fusion of left and right hands)	2x3	Two weeks to three years	Left and Right	IP: 99.67 % VP: 0.33 % EER

mance operation;

- The algorithm can accept input from imaging devices with as low a resolution as 30 dpi and hands containing various accessories;
- The hand normalization system can work with a wide range of acquisition devices such as scanners and low-resolution cameras;
- Hand biometric access control can be applied very reliably to populations from hundreds to a thousand subjects;
- The hand-biometric system trained on a given population can be exported to operate on a partially or totally differing population;
- The algorithm does not suffer noticeable performance loss over time lapses from several months to a year.

4. 3D FACE RECOGNITION

4.1. Introduction

Automatic identification and verification of humans using facial information is one of the most active research areas in computer vision. Due to the wide use of digital cameras and ease of the acquisition, the main effort is put on the recognition of faces from 2D intensity images. However, there are a number of challenges encountered with face recognition from 2D intensity images. In intensity images, faces acquired from the same person show high variability due to lighting conditions. Face segmentation from a cluttered background is another unsolved problem.

The shape information of 3D faces is descriptive enough to distinguish people. This information can either be used alone, or can be fused with 2D intensity information to increase recognition performance. Three-dimensional face recognition possesses certain benefits over intensity-based 2D face recognition: The two crucial advantages are the illumination-invariance and the ease of detection and cropping of the face region from the background. Since 3D acquisition devices measure shape information, 3D face models are independent of lighting conditions. Segmentation of 3D faces from background is relatively an easy task for range images, as far as the face is within the range of the scanner. Furthermore, 3D face information can model small pose variations as opposed to intensity images. Due to these advantages of 3D face based biometry and due to the advancements in 3D scanning technologies, there has been a rapid increase in research efforts on 3D face biometry in the last decade [92].

Expression variation remains as a challenge for 3D face recognition systems [93]. This point is illustrated in Figure 4.1, where we show face scans of three subjects, each with three varying facial expressions, from the Face Recognition Grand Challenge (FRGC) database [94]. We will propose solutions to expression variation in Chapter 5.

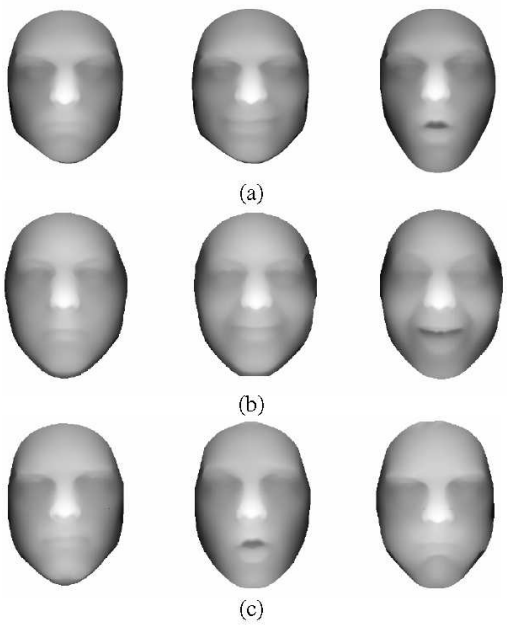


Figure 4.1. 3D faces from three different subjects (a, b, c). Faces on the same row correspond to the same person with different facial expressions.

The high quality range scans of 3D faces contain hundreds of thousands of dense points. This high dimensional representation makes the matching stage inefficient, especially for real time applications. Since subspace methods are excellent dimension reduction techniques, we propose to use them for feature extraction. The 3D face-based biometry is a relatively new research area, therefore many conventional signal processing and subspace extraction techniques were not considered yet. Some of the subspace-based feature extraction schemes (DFT, DCT, ICA and NMF) were not previously applied to 3D face representations.

4.2. Previous Work on 3D Face Recognition

Point-cloud representation is one of the popular representations in 3D face recognition community [95, 96]. The point-cloud representation of a probe face is registered to the gallery faces by the Iterative Closest Point (ICP) method. The quality of the ICP alignment is supposed to be sufficiently good to allow for pointwise matching of two face point clouds. In [97] and [98], all the point sets of the probe and

gallery faces are registered to an average face via the ICP in order to align the faces to a common reference frame and to establish dense correspondences. Then, the features are extracted from these aligned point sets. We also followed this scheme in our work, and extracted our subspace based features from the aligned point sets.

There are several alternatives to the ICP-based matchers. Koudelka et al. [99] automatically find several facial landmarks such as nose tip, sellion, inner eye corners, and mouth center and then sample a number of random points in their neighborhood. They use a combination of ICP and Hausdorff algorithms to match two facial surfaces.

Instead of rigid registration via ICP, nonrigid versions of it can be beneficial in establishing the correspondence between facial surfaces. For example, Irfanoglu et al. [97] propose the thin-plate-spline (TPS)-warping algorithm. First, they automatically locate several facial landmarks, and then warp a given face image to an average face model (AFM) using TPS. Passalis et al. [100] propose a generic face model, which is fitted to a given face. The related displacement information forms a separate deformation image. The authors perform wavelet analysis on this deformation image to get the descriptors.

A number of algorithms were proposed to deal with the deformation of the geometric structure of the face due to expression. One approach is to model the face as a deformable object. Lu and Jain [101] have suggested the use of person specific deformable models. The deformations are learned from a small group of subjects. Then, the learned deformation model is transferred to the 3D neutral model of each subject in the database via TPS. At the matching stage, the person-specific deformable models are fitted to the test face using a modified ICP algorithm where deformation parameters are updated in an iterative way.

Besides ICP, there are other schemes where the registration [102] or correspondence matching process [103, 104] is inherent to the recognition algorithm. Mian et al. [103, 104] used rotation invariant tensors that are constructed in locally defined

coordinate bases to represent the 3D faces. At the recognition stage, the best matching pairs of features, i.e., the correspondences, between the template and test images are found either by exhaustive matching [104] or via a 4-D hash table [103]. Bronstein et al. [102] proposed an expression-invariant face recognition algorithm, where one 3D face is embedded onto another face by multidimensional scaling (MDS). The MDS is used to establish intrinsic geometric correspondence between two similar but deformed surfaces.

Another approach to deal with expression variations is to adopt a region-based scheme. Chang et al. [105] use three overlapping face regions around the nose. These regions are assumed to be less deformable under expressions as compared to those facial parts including eyes and mouth. The corresponding facial region pairs from the gallery and probe images are matched with Iterative Closest Point (ICP) algorithm, and the matching scores are combined with the product rule. Any other region that is deemed deformable under expressions is ignored. Faltemier et al. [106] describe a system, where one pre-determined facial region in the gallery image is compared with multiple regions in the probe image, and then their outcomes are combined through committee voting. A more general treatment of local region-based face recognition system is presented in [107] and [108] for 2D and 3D face modalities, respectively. The underlying principle is the automatic determination of discriminative parts of facial regions via feature subset selection heuristics. These authors show that, even without prior knowledge on the importance of facial subregions, one can learn informative facial parts from the data, which leads eventually to better identification rates.

Samir et al. [109] represented a facial surface as a collection of planar curves derived from the level sets or from the geodesic curves that are centered at the nose tip of the face. The second type of representation is based on geodesic curves and is invariant to rotation.

Another popular approach in 3D face recognition research is to convert the 3D point-cloud information into 2D depth images (range images). While the 2D

data are more familiar to work with, the loss of intrinsic face information due to resampling and mapping to a regular grid must be accounted for. When more than one point is mapped to a cell in a 2D grid, these points are undersampled during a conversion to 2D. A case in point is the sloping parts of the face, which suffer due to the foreshortening effect in the 3D to 2D conversion. Some of these sloping parts may incorporate interperson differences like the slopes of nostrils. Once the depth image is formed, one can treat the 3D face recognition problem as simply a 2D image matching problem.

Pan et al. [110] design a pose-invariant recognition system by projecting the preregistered 3D point cloud data to a plane parallel to the face plane. They achieve pose invariance via a variant of the ICP-based registration. Their projection flattens out the facial surface. Then they apply PCA to extract features.

An approach for matching range images, using the original measured data and not their subspace projection, is discussed in [111]. In that work, Russ et al. apply the partial-shape Hausdorff distance metric to range images. The motivation behind using the Hausdorff distance is its partial invariance to inconsistencies such as noise, holes, and occlusions in the 3D facial data.

As an alternative to depth images, it is also possible to construct 2D images that represent other properties of 3D data, such as surface curvature and surface normals. Abate et al. [112] generate normal maps, which store three-variate mesh normals in lieu of the red, green, and blue (RGB) components. The difference between the normal maps of the two images is calculated in terms of three difference-angle histograms.

There are a number of papers concentrating on local surface features such as curvatures. Tanaka et al. [113] utilized Extended Gaussian Image, which includes information of principal curvatures and their directions. Different EGIs are compared using Fishers spherical correlation. Another work based on Extended Gaussian Image can be found in the paper of Lee et al [114]. Gordon [115] proposed

a template-based recognition system, which again involves curvature calculation. Chua et al. [116] have used point signatures, a free form surface representation technique. Beumier et al. [117] extracted central and lateral face profiles, and compared curvature values along these profiles.

4.3. Types of Face Representation

We assume we have registered 3D coordinate data coming from the preprocessing stage. The common approach for registration is alignment of the 3D point cloud of a probe image onto each gallery image. Since ICP is a time-consuming procedure, the alignment of an input face to all the faces in the database precludes real-time operation. Therefore we follow the Average Face Model (AFM) approach introduced by Irfanoglu et al. [97]. The AFM is obtained from a set of training face samples. Then the 3D point cloud of the probe and gallery faces are aligned to AFM via ICP. This scheme allows us to rapidly build correspondences among faces. The details of the preprocessing and alignment stages can be found in [92].

The 3D face data admit various representation styles with their consequent extracted features. We use three different representation schemes for recognition: Point cloud, depth image and 3D voxel representation. In the following sections, we briefly describe the construction of each representation. Point cloud and depth image representations are common in 3D face recognition research. However, mapping the 3D point cloud onto a voxel grid is new in the literature.

4.3.1. Point Cloud Representation

The point cloud is the set of the 3D coordinates (x, y, z), of the points of a face object. A face with N samples is simply represented in terms of three coordinate vectors, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ of length N. Figure 4.2a shows a sample point cloud, and Figures 4.2b, c and d show the three coordinate vectors plotted with respect to the vector index. Notice that all correspondences among points of different faces must have been determined at the registration step.

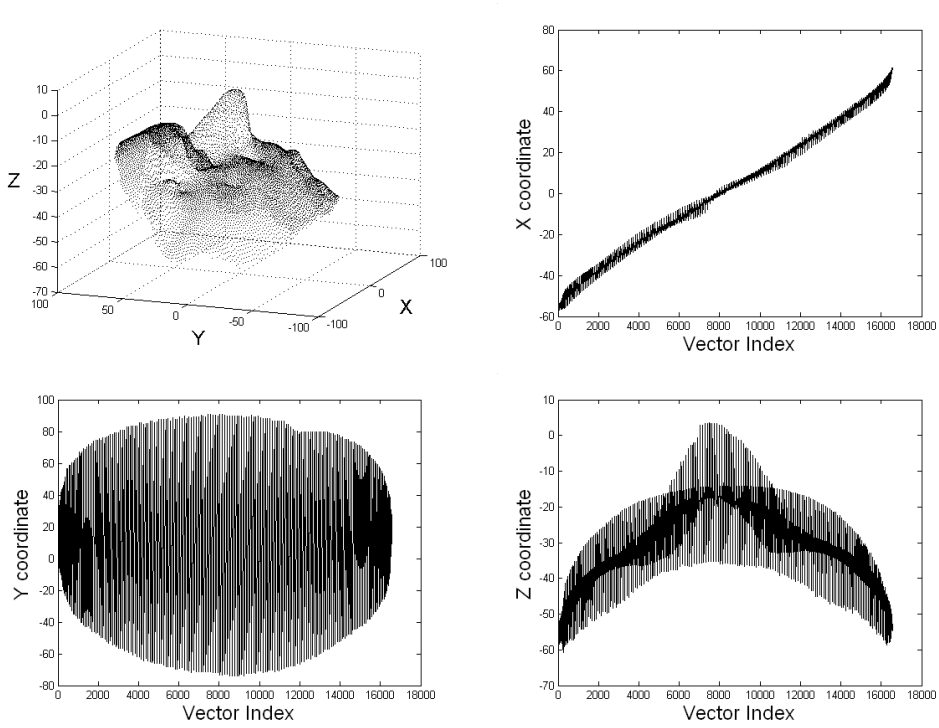


Figure 4.2. (a) Point cloud representation. (b, c, d) X, Y, Z coordinate vectors respectively, as a function of the vector index.

Although the ensemble of face point encodes the variations among different faces, there is a very loose neighborhood information in the point cloud representation due to the one dimensional vector structure of the coordinates. The simplest scheme is to use the coordinates themselves as features and calculate the sum of Euclidean distances between corresponding points of two faces. We propose to apply subspace-based techniques directly to the point cloud as described in Section 4.4.

4.3.2. Depth Image

One of the most conventional ways to represent face data is the depth image where the z-coordinates of the face points are mapped on a regular x-y grid by using linear interpolation. The depth image has the form of a 2D function $I(x, y)$, similar to an intensity image (Figure 4.3.2). Thus many techniques applicable to intensity images for classifying facial appearance variations can be directly used

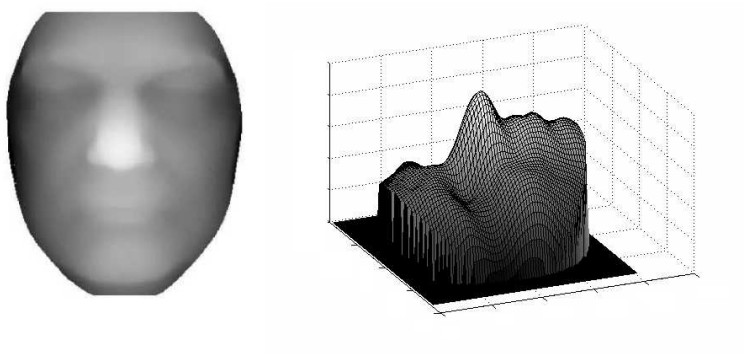


Figure 4.3. 2D depth image from side and from top.

for depth images to bring forth facial landscape differences among subjects. The classical dimensionality reduction techniques such as PCA, LDA, and ICA have been previously applied to depth images [98, 118, 119, 120]. In Section 4.4, we consider a number of feature extraction techniques applicable to depth images.

4.3.3. 3D Voxel Representation

The initial point cloud data can be converted to a voxel structure, denoted as $V_d(x, y, z)$, by imposing a lattice. The first step of the voxel conversion procedure is to define an $N \times N \times N$ grid box in such a way that the barycenter of the point cloud coincides with the center of the box. Then, we define a binary voxel occupancy function $V(x, y, z)$ on this grid. This is simply an indicator function: if, in a cell at location (x, y, z), there does not exist any points of the cloud, $V(x, y, z)$ is set to zero. If there are one or more points in that cell, then the binary function at that voxel location assumes the value one. Therefore all cells on the face have the value of one and the rest of the cells in the space are set to zero, which, in effect, defines a 3D shell. Figure 4.4 shows a sample point cloud, and the corresponding 3D binary function, $V(x, y, z)$, displayed as a negative image.

We have found advantageous to convert the binary voxel data into continuous form via the distance transformation. We apply 3D distance transform to the binary function $V(x, y, z)$ to fill the voxel grid and obtain $V_d(x, y, z)$. The distance transform is defined as the smallest Manhattan distance of a voxel point to the binary surface.

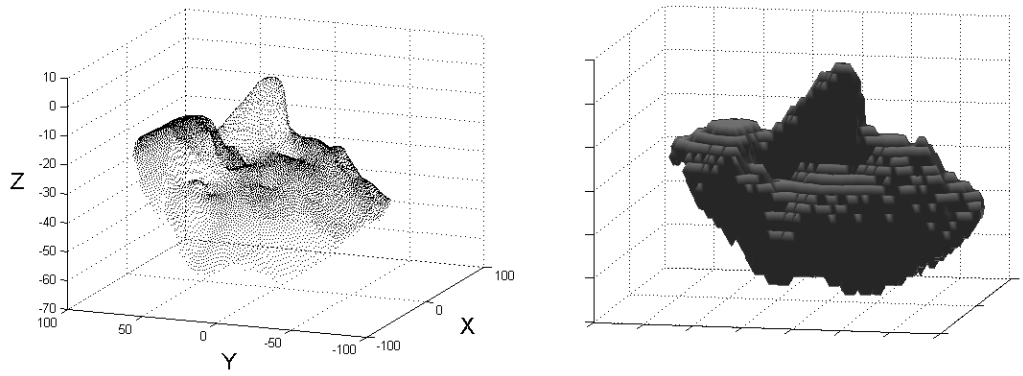


Figure 4.4. The point cloud and its binary voxel representation.

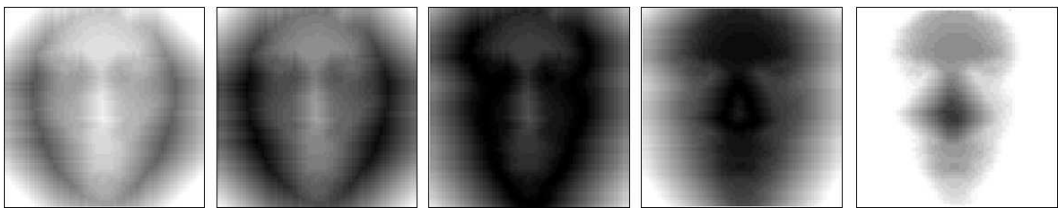


Figure 4.5. Slices from the voxel representation based on the distance transform.

This function gets a value of zero on the face surface, and it increases as we go further away from the surface. By using the distance transform, we distribute the shape information of the surface throughout the 3D space and obtain a smoother representation compared to the binary voxel description. Figure 4.5 gives slices from the voxel representation based on the distance transform.

4.4. Facial Feature Extraction Methods

We have explored a set of subspace-based features that extract discriminative information from 3D faces. We have a number of combinations of the representation types and feature extraction methods. For example, DFT was applied on the voxel representation and on the depth image. Similarly, ICA was applied to the point cloud and depth field representations of 3D faces. The combinations we have tested can be seen in Table 4.1. We assume that 2D data (e.g., depth images, intensity

Table 4.1. Representation schemes and features used for 3D face recognition.

Representation	Features
3D Point Cloud	2D DFT
	ICA
	NMF
2D Depth Image	Global DFT
	Global DCT
	Block-based DCT (Fusion at feature level)
	Block-based DCT (Fusion at feature level)
	Block-based DFT (Fusion at decision level)
	Block-based DCT (Fusion at decision level)
	ICA
	NMF
3D Voxel Representation	3D DFT

images) have size $N = N_1 \times N_2$, the point clouds have size $N_p = N \times 3$ and that 3D voxel data have size $N = R \times R \times R$.

4.4.1. DFT and DCT on 3D Face

We have employed DFT-based features for both the 3D point clouds, for the depth images and for 3D voxel data. The point cloud and depth image representations provide neighborhood information. In the point cloud representation, the ordering of the points only provides point-to-point neighborhood. However DFT/DCT coefficients are highly dependent on the spatial arrangement of the signal points.

2D-DFT of point clouds: In order to compute 2D-DFT coefficients from the point cloud of N_p points, we first define an $N_p \times 3$ matrix $\mathbf{P}$, where we put the $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ coordinates of the N_p points into the columns: $\mathbf{P} = [\mathbf{X}\ \mathbf{Y}\ \mathbf{Z}]$. We apply 2D DFT on this 2D matrix. We could have concatenated the $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ coordinates and computed

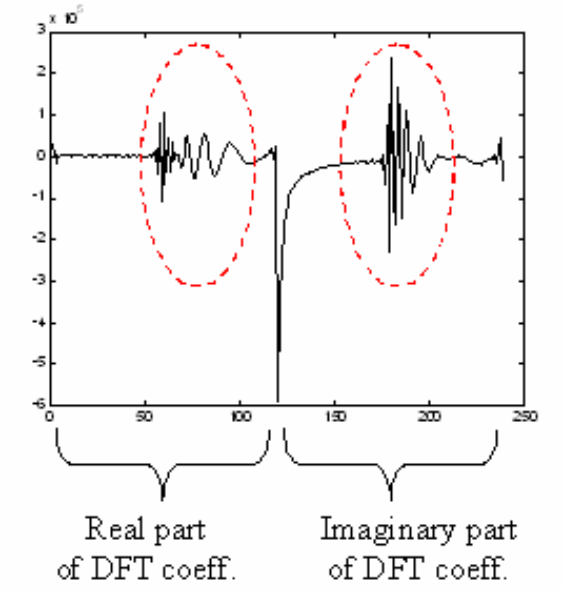


Figure 4.6. Sample DFT-based feature vector obtained from point cloud.

the one-dimensional DFT, however, then we would lose the inherent relation within the coordinates of a point. DFT coefficients are strongly dependent on the order of the data, and we intended to keep the **X**, **Y** and **Z** coordinates of a point, close in the data structure. The 2D-DFT coefficients of **P** are then computed as follows:

$$\mathbf{FP}_{uv} = DFT\{\mathbf{P}\}_{uv} = \sum_{n=1}^{N_p}\sum_{d=1}^{3} \exp\left(-\frac{2\pi nu}{N}\right)\exp\left(-\frac{2\pi dv}{3}\right)\mathbf{P}_{nd} \tag{4.1}$$

FP is a matrix of size $N_p \times 3$. We take the first K coefficients of the first column of this matrix, and obtain a feature vector of size $2K - 1$ by concatenating the real and imaginary parts of the K complex coefficients. Figure 4.6 shows a sample DFT-based feature vector of the point cloud. One should note that, most of the energy is concentrated in the band-pass region due to the zigzag scan of the face as can be observed from the plots of the coordinates in Figure 4.2.

4.4.1.1. Global 2D-DFT and 2D-DCT of Depth Images. For a depth image $I(x, y)$ we calculate its $N_1 \times N_2$-point DFT and extract $K \times K$ low-frequency coefficients to form a feature vector of size $2K^2 - 1$, by concatenating the real and imaginary parts of the

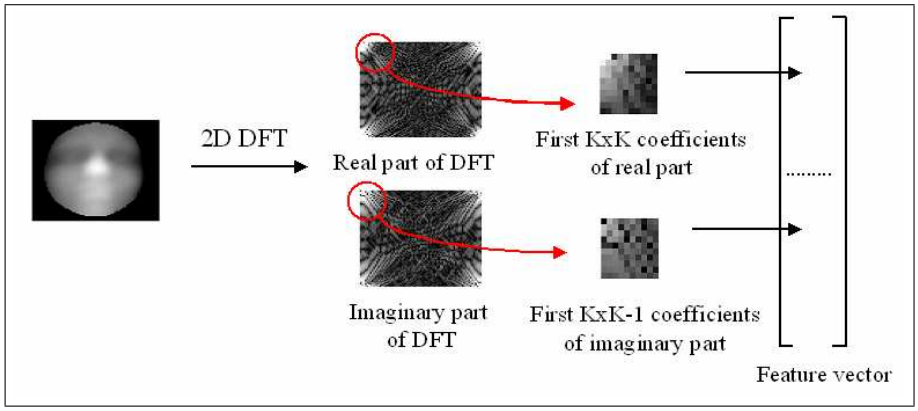


Figure 4.7. Extraction of global DFT-based features from depth image.

coefficients (Figure 4.7). Likewise, we compute the global DCT: However, in this case, we obtain a feature vector of size K^2 since DCT coefficients are real.

4.4.1.2. Block Based 2D-DFT and 2D-DCT of Depth Images. In addition to the global DFT/DCT-based techniques, we also extract local features, based on the calculation of DFT coefficients on blocks. The depth images are partitioned into blocks of size $M \times M$ and 2D-DFT is applied separately to each block. Then we take the first $K \times K$ DFT coefficients to form the feature vector special to a particular block. We can then fuse this data either at feature level, or at decision level.

Fusion at feature level is performed by concatenating the DFT coefficients coming from the blocks in a single vector. Figure 4.8 explains the procedure.

We perform fusion at decision level by using the sum rule. The depth image of an input face to be recognized is partitioned into blocks and each block is matched with the corresponding blocks of the depth images in the database. From this comparison, each face in the database gets a rank. A face in the database, thus obtains rank values as many as the number of blocks. When we sum up the ranks, we obtain the final rank for the face, and choose the identity of the face with the lowest final rank. Figure 4.9 summarizes this procedure.

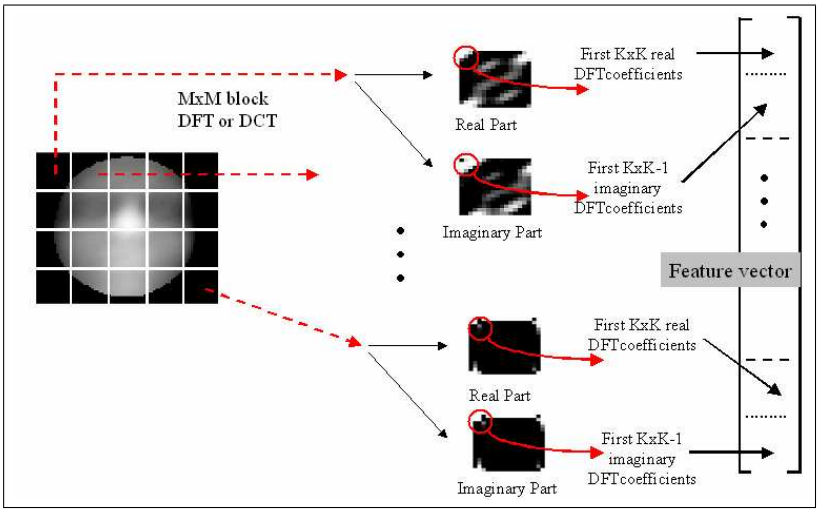


Figure 4.8. Procedure for fusion at feature level.

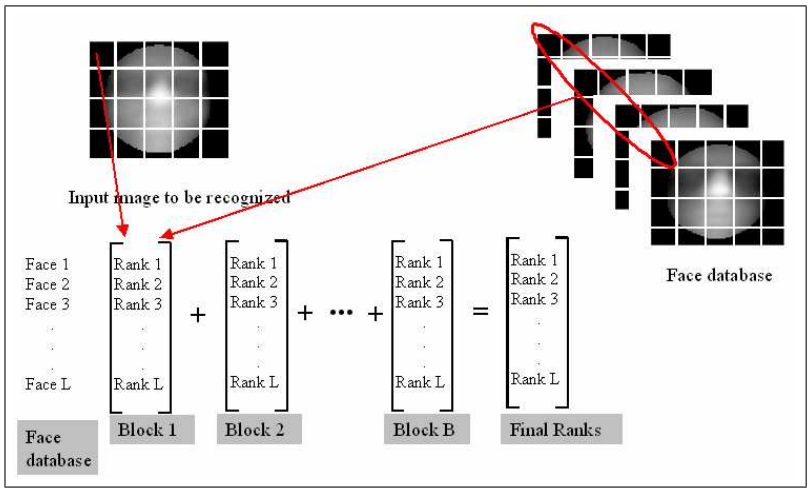


Figure 4.9. Procedure for fusion at decision level.

4.4.1.3. Global 3D-DFT of Voxel Representation. For faces represented in terms of voxels, we compute the 3D-DFT of its distance transform $V_d(x, y, z)$. The feature vector of size $2K^3 - 1$ is obtained by concatenating the low-pass $K \times K \times K$ real and imaginary terms, as shown in Figure 4.10.

4.4.1.4. Matching DFT/DCT Coefficients. Faces have typically slowly varying surface characteristics, which means that there exists a rapid power differential in DFT/DCT coefficients with increasing frequency. We only select the $K \times K$ ($K \times K \times K$ for the 3D voxel data) low-pass coefficients, where K is no larger than 10. While the energetic coefficients at DC and at very low frequencies represent the gross structure, a portion of the higher frequency coefficients carry the shape difference information between individuals. These coefficients, which are important for face classification, tend to be eclipsed by the heavy-weight coefficients. This problem can be remedied by the QR-decomposition technique. We thus apply QR-decomposition to these feature vectors: $\mathbf{F} = \mathbf{QR}$ where $\mathbf{F}$ is the matrix consisting of feature vectors if we have only one training sample per individual. For the case of more than one sample per individual, $\mathbf{F}$ contains the difference of the feature vector of each subject to its class mean. In this case the QR-decomposition corresponds to a variant of linear discriminant analysis, where $\mathbf{F}$ corresponds to the within class scatter matrix. $\mathbf{R}$ is the upper triangular matrix obtained from QR-decomposition of the training features. In effect, we transform all feature vectors in both training and test sets by multiplying them with the inverse of $\mathbf{R}$, so that a feature vector f is mapped to $f^T \leftarrow f^T\mathbf{R}^{-1}$. Finally, the transformed test and training feature vectors are compared using the cosine distance.

4.4.2. ICA on 3D Face

We test the potential of the ICA scheme as a discriminative feature for 3D face data. We extract ICA coefficients from either the 3D point cloud or the depth image representation.

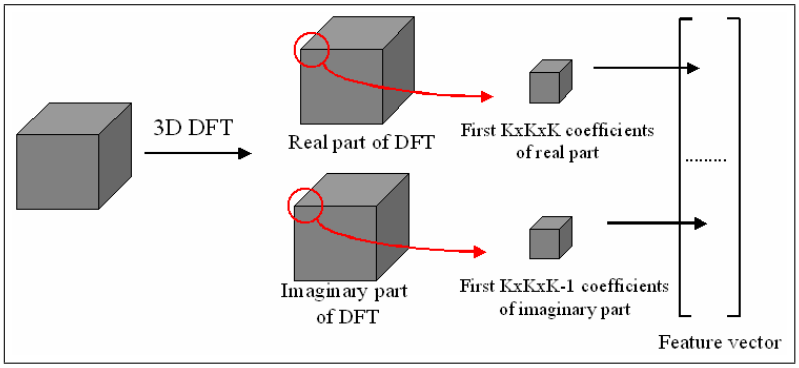


Figure 4.10. Extraction of global DFT-based features from voxel representation.

For the point cloud all x, y and z coordinates of a face are concatenated to a single vector. Its dimensionality is then reduced by applying PCA to the training set of point-cloud vectors. The columns of the data matrix **X** for the ICA analysis are constituted of the first *K* PCA coefficients of the faces. Then, the FastICA algorithm described by [20] is applied to obtain the basis **A** and the independent coefficients **S**. Finally, we apply QR-decomposition technique to the ICA-based features to re-weight the elements of the feature vector according to their discriminative power.

The ICA analysis for depth images follows a similar procedure. The columns of a depth image are concatenated to form a single one-dimensional vector, one for each face. This data is subjected to PCA reduction, ICA decomposition and QR normalization.

Figure 4.11 shows the first 10 basis functions derived from PCA, whereas Figure 4.12 shows 10 independent face components. PCA only captures the second order variations due to the general face geometry, while ICA faces represent individual faces within the database fairly well. One can observe more face-like structures from the ICA basis images.

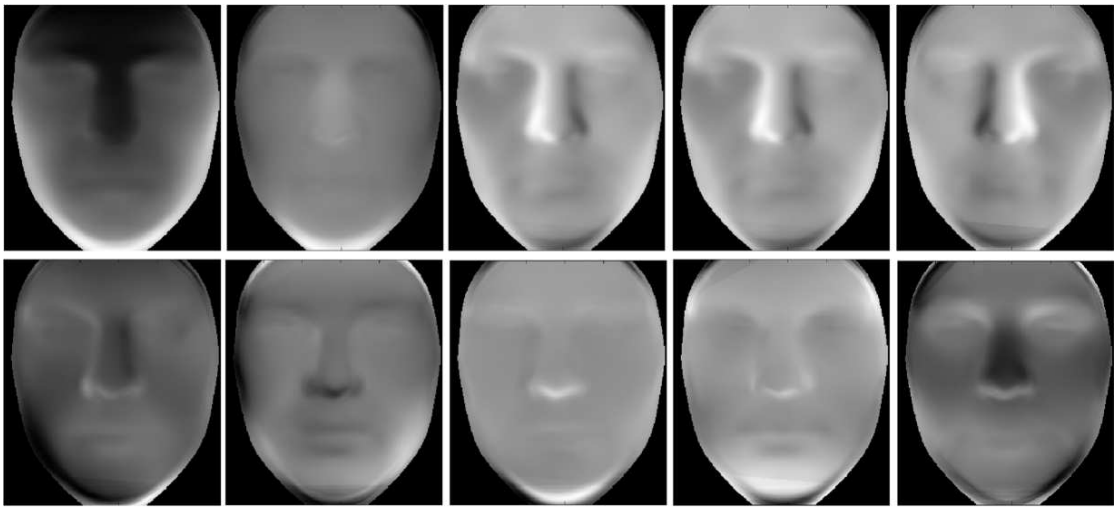


Figure 4.11. First 10 basis faces obtained from PCA applied on depth images.

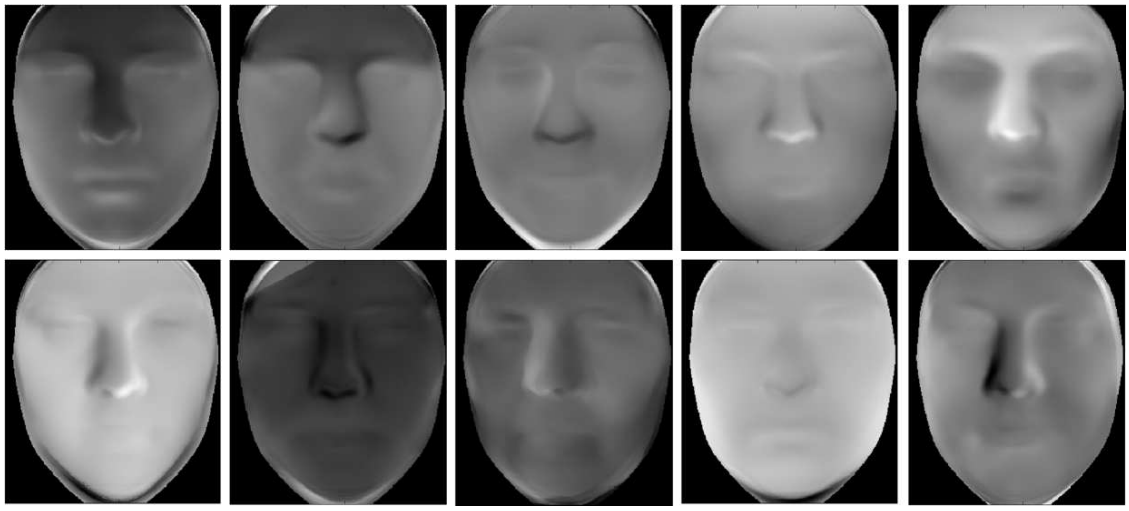


Figure 4.12. Basis faces from ICA of depth images.

4.4.3. NMF on 3D Face

Parallel to the preprocessing stage of ICA decomposition, we first apply PCA to reduce the dimensionality of the raw data (depth or point cloud information) and place the first M PCA coefficients of each face into the columns of the data matrix. We add a constant to the PCA coefficients to obtain a nonnegative data matrix. The nonnegative factors $\mathbf{V}$ and $\mathbf{H}$ are obtained using the multiplicative update rules described in [23]. Then the QR-decomposition is applied to the NMF-based features as described in Section 4.4.1.

4.5. Experimental Results

4.5.1. Results on the 3D-RMA Database

The 3D-RMA database [121] contains face scans of 106 subjects. The total number of faces is 617 and there are five to six sessions per person. We have used four sessions for training (424 face scans) and utilized the rest 193 faces for testing. We have conducted five experiments by selecting different combinations of the sessions. Table 4.2 gives the identification performances (IP) of all the schemes, averaged over the five experiments. Table 4.2 also provides information about the number of features selected for each scheme.

As can be observed from Table 4.2, ICA and NMF-based features extracted from the point cloud representations of faces gave superior results with smallest number of features. They gave 100 per cent recognition performance for the three experiments and missed only one face for the two experiments. The missed face is plotted on top of another face of the same person in Figure 4.13. The misclassification is due to the inaccurate registration of this particular face.

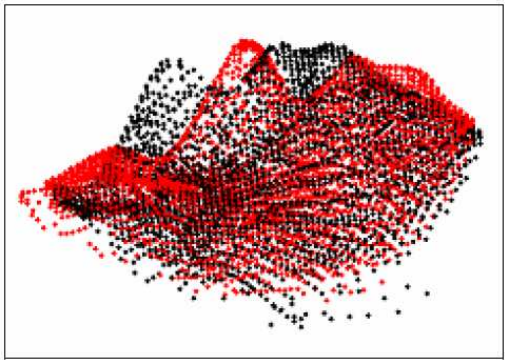


Figure 4.13. The misclassified face plotted on top of another face of the same person (misclassified by ICA and NMF based features computed on point cloud representation).

4.5.2. Results on the FRGC v1.0 Database

For the recognition tests, we have used the University of Notre Dame (UND) 3D face database [94], also known as the Face Recognition Grand Challenge (FRGC) v1.0 database in the literature. The original UND database contains 943 3D scans of 275 subjects. We had to use a subset of the original database, since 75 subjects had only one scan, and 14 3D scans were badly registered with the texture data. Thus, the part of the database involved in our experiments contained 854 2D and 3D scans of 195 subjects. Each subject had at least two, and at most eight 3D scans. The UND database consists mostly of frontal faces and does not exhibit significant expression variations. However, some scans have slight in-depth pose variations, and different expressions. Shape data contain approximately 30,000 - 40,000 3D coordinates.

We have designed four different experimental configurations, as shown in Table 4.3. Each configuration contains a different number of training samples per subject. The subscript i in experiment E_i denotes the number of training samples per subject in that experiment. The reason for different populations is that in the UND database, 195 subjects have more than two 3D scans, 164 subjects have more than three scans etc. Thus E_1 is designed so that every subject possesses only one image in the training set, and while the rest of $854 - 195 = 659$ images are placed in the

Table 4.2. Recognition performances on the 3D-RMA database.

Representation	Features	Number of features	IP (%)
3D Point Cloud	2D DFT	2x400-1 (799)	95.86
	ICA	50	99.79
	NMF	50	99.79
2D Depth Image	Global DFT	2x8x8-1 (127)	98.24
	Global DCT	11x11 (121)	96.58
	Block-based DCT (Fusion at feature level)	20x20 blocks (12 blocks), 2x2x2-1 for each block (84)	98.76
	Block-based DCT (Fusion at feature level)	20x20 blocks (12 blocks), 3x3 for each block (108)	98.24
	Block-based DFT (Fusion at decision level)	20x20 blocks (12 blocks), 4x4-1 for each block (180)	98.13
	Block-based DCT (Fusion at decision level)	20x20 blocks (12 blocks), 6x6 for each block (432)	97.82
	ICA	50	96.79
	NMF	50	94.43
3D Voxel	3D DFT	2x4x4x4-1 (127)	98.34

test set. For each experiment, we have run several folds, and the number of folds for each experiment is shown in the last column of Table 4.3. We report only the average of the recognition accuracies of the folds. The most difficult experiment is obviously E_1 (single gallery experiment) since not only there exists a single training image per person, but also both the enrollment size and the number of test scans are larger. Conversely, the easiest experiment is E_4, since it contains four training images per person and the test size is smaller. We choose not to report the even easier identification experiments, such as E_5 and E_6, since they are not sufficiently challenging. Note that when the number of images in the training set increases, the number of subjects that participate in that experiment decreases.

In order to provide a more complete evaluation of our subspace-based methods, we have included the surface-based methods proposed in [122] in our comparisons. First is the surface normals-based matching, where the surface normals of corresponding points of two faces are compared with L_2 norm and summed up. There

Table 4.3. Experimental configurations for FRGC v1.0.

	Training samples per subject	Number of Subjects	Total training scans	Total test scans	Fold count
E_1	1	195	195	659	2
E_2	2	164	328	464	3
E_3	3	118	354	300	4
E_4	4	85	340	182	5

are four methods based on the curvature of the points on the face surface, namely, shape index, Gaussian curvature, Mean curvature and principal directions. We also include the point difference methods for the coordinates of the 3D point cloud and the depth values of the range images.

Table 4.4 shows the recognition performances on FRGC v1.0 database for the four experimental setups. There is a jump difference in performance between single gallery case and the experiments with at least two training images per subject. This result means that all of the subspace methods provide class separable features. When we have four gallery images per subject the performance is over 99 per cent for all the methods. With NMF we achieve 100 % correct classification. We conjecture that the subspace techniques achieve their full potential when adequate training data are supplied to construct their feature subspaces. The subspace techniques need more training samples to model the within class variability through the analysis into basis faces and the corresponding coefficients. The final QR normalization step in the subspace-based techniques also require at least two training samples per subject in order to reweight the features according to class separability.

The depth-image-based classifiers DI-ICA and DI-NMF obtain 72 per cent average performance rate. On the other hand, with the point-cloud representation, PC-ICA and PC-NMF achieve 85 per cent average recognition rate. Hence, it is the representation (depth versus point cloud), rather than the feature extraction tool (ICA versus NMF), that is the determining factor. As matrix factorization techniques, ICA and NMF give similar results on the same representation.

Table 4.4. Recognition performances in percentages on the FRGC v1.0.

Representation	Features	Number of features	E_1	E_2	E_3	E_4
3D Point Cloud	Point coordinates	49,680	87.71	94.68	97.92	98.90
	NMF	90	85.13	97.77	99.25	100.00
	ICA	90	85.66	98.71	99.67	99.89
2D Depth Image	Depth values	90,201	55.99	70.19	79.75	87.69
	DCT	49	78.53	97.63	99.58	99.78
	DFT	49	75.95	97.13	99.08	99.56
	ICA	80	72.46	96.55	98.92	99.01
	NMF	70	71.55	95.83	98.67	99.67
Voxel	DFT	53	64.26	91.16	97.92	99.34
Surface Normal	Surface normals	49,680	89.07	96.84	98.92	99.45
Curvature	Shape index (SI)	16,560	90.06	96.55	98.67	99.34
	Principal Directions (PD)	99,360	91.88	97.13	99.08	99.45
	Mean Curvature (H)	16,560	87.41	95.69	98.50	98.90
	Gaussian Curvature (K)	16,560	84.37	93.89	97.25	98.46

4.5.3. Results on the FRGC v2.0 Database

In the experiments with FRGC v2.0, we have only considered the case where there is only one gallery image in the database. Since we have called the corresponding experiment protocol E1 for the FRGC v1.0 data set, we call this protocol E1. However, the two experimental protocols have an important difference: we have used the FRGC v1.0 to train our subspace-based methods such as the ICA and NMF and used the class information in FRGC v1.0 to estimate the LDA and QR normalization parameters. Then, these parameters and the basis images were fixed and were used to calculate the feature vectors of the gallery images as well as the probe images of FRGC v2.0. We have chosen the earliest scan of each subject as the gallery image. All the 410 gallery images are neutral, i.e., they do not have facial expressions. All the other scans are used as test images: Thus, we have 3542 test images. Some 1984 of the test images are neutral faces, and the remaining 1558 faces exhibit expression variations.

Table 4.5. Recognition performances in percentages on the FRGC v2.0.

Representation	Features	Number of features	E_1
3D Point Cloud	Point coordinates	70	80.07
	NMF	200	86.34
	ICA	300	88.31
2D Depth Image	Depth values	600	57.82
	DCT	169	76.14
	DFT	127	73.97
	ICA	450	67.25
	NMF	300	62.68
Voxel	DFT	127	72.67
Surface Normal	Surface normals	50	83.79
Curvature	Shape index (SI)	80	75.30
	Principal Directions (PD)	85	80.35
	Mean Curvature (H)	80	72.56
	Gaussian Curvature (K)	80	70.78

Table 4.5 shows the classification rates on FRGC v2.0 with the single-gallery-image setup. The FRGC v1.0 database has been used to tune the parameters of the subspace-based methods, the QR normalization, and the linear discriminant functions. In the FRGC v2.0 experiments, we apply the LDA to the features of the coordinates of the point cloud, to the depth values, to the surface normals, and to the curvature based methods.We apply LDA and PCA to these raw feature vectors in the v2.0 database. Our experimental results show that, with the help of FRGC v1.0 training set, it is possible to significantly improve the identification rates of these methods when compared to using their raw features only.

The second column of Table 4.5 displays the feature dimensionality of each method. For the methods that use LDA or PCA, dissimilarities between feature vectors in the transformed subspace are calculated using the cosine distance. For DFT-, DCT-, ICA-, and NMF-based methods, we have increased the dimensionality in subspace-based techniques (when compared to the FRGC v1.0 experiments) since we need more features to discriminate between the subjects in a larger database. The

individual performances in FRGC v2.0 can be interpreted in relation to the results obtained with FRGC v1.0 as follows:

- Point-cloud-based ICA and NMF methods perform best, yielding 88.31 per cent and 86.34 per cent identification accuracies, respectively. Since we have built the subspace models using FRGC v1.0, we had enough data to construct the subspaces.
- In general, point-cloud-based methods perform better than depth-image-based methods. The best depth-image based method, namely, the DCT methods, reaches 76.14 per cent identification rate, whereas all of the point-cloud approaches attain identification rates greater than 80 per cent.
- The best two surface-descriptor-based approaches, the surface normals and the principal directions, attain 83.79 per cent and 80.35 per cent recognition rates, respectively.

Since the subspace based methods (ICA, NMF, DFT, DCT) and the local descriptors (Surface normals, curvatures) give different descriptions of the faces, fusion of these methods improves the classification rates beyond individual methods.

Table 4.6 shows the performance improvement due to fusion of 16 different face experts in single gallery experiment of FRGC v2.0. In addition to the 14 methods listed in Table 4.5, two texture-based methods are included (Gabor features and raw pixel values). The fixed combination rules, the sum and product rules obtain 93.56 per cent and 93.08 per cent identification rates, which are 5.25 per cent and 4.77 per cent better, respectively, than the best individual face expert (ICA on point cloud). The last row of Table 4.6 gives the fusion result with the subset of the experts, selected by Sequential Floating Backward Search (SFBS) method [122]. This method selected the set of the following seven methods as the best performing subset of all the 16 experts: (i) ICA on point clouds, (ii) DCT on depth images, (iii) point cloud coordinates, (iv) surface normals, (v) principal directions, (vi) raw texture pixels, and (vii) Gabor features of texture. These seven classifiers attain 95.45 per cent identification rate which is 7.14 per cent better than the best single face expert.

Table 4.6. Recognition performances in percentages with fusion on the FRGC v2.0.

Fusion method	Fused experts	Recognition rate	Improvement
Best method	ICA on points cloud	88.31	
SUM rule	All	93.56	5.25
PRODUCT rule	All	93.08	4.77
SUM (SFBS selection)	7 experts	95.45	7.14

Table 4.7. Recognition performances in the literature on the FRGC v2.0.

Reference	Number of gallery faces	Number of probe faces	Landmarking scheme	Recognition rate
Passalis et al. [100]	466	3541	Automatic	89.5
Chang et al. [105]	449	3939	Automatic	91.9
Chang et al. [105]	449	3939	Manual	92.9
Faltemier et al. [106]	410	3541	Automatic	94.9
Our methods (SBFS selection)	410	3542	Manual	95.45

4.5.4. Comparison with the State of the Art

Table 4.7 illustrates the performances of different algorithms in the literature, which use FRGC v2.0 for identification simulations. In all of these systems, the performance of the proposed approach is benchmarked via single-gallery experiments where the earliest scans of each subject are placed into the gallery set. However, the experimental setups are different with different sizes of gallery and probe sets. In this respect, the experimental protocol used by Chang et al. [23] is more challenging since they conducted recognition experiments on a larger database spanning both FRGC v1.0 and v2.0 image sets. Furthermore, their results are obtained via a fully automatic face recognition system, whereas our system employs manual landmarking for registration. Thus, the performance figures should be compared with respect to the relative difficulty of each experimental setup.

4.6. Conclusions

We have designed a diverse set of 3D face recognizers that differ in the face representation and/or in the discriminative features they extract from these representations. We have conducted our experiments on the FRGC v1.0 and v2.0 data sets. We have used the experimental configurations used by the most recent studies. In the experiments on the FRGC v1.0 data set, we have used all experimental configurations Ei. However, in FRGC v2.0, we restricted our attention to E1 experiments, where the gallery contains a single training image per subject. In the experiments with FRGC v2.0, we have used the FRGC v1.0 data set to learn feature subspaces and selections for expert consultations. We have conducted extensive experiments on the effectiveness of different features, different representations, and different fusion rules. By experimenting with different training-set sizes, we were able to draw conclusions on the effect of training sets.

Representation is more important when training set is small. The acquired face data in 3D can assume one of the forms of point clouds, surface normals, depth images, curvatures, or 3D voxels. The depth image derived from the original 3D face is also treated as a 2D image. In experiments where the training-set size is very small, the effect of representation type dominates.

The effect of matching feature dominates when training set size gets larger. The second tier of the analysis is the feature extraction stage. For 3D face data, we have compared two varieties of features, namely, the subspace features (DFT, DCT, NMF, and ICA) and the spatial geometric features (point cloud, shape index, surface normals, and principal curvatures). Subspace-based methods such as application of ICA and NMF on point clouds gave superior results when a large training set is available. One important conclusion is that all 3D face representation types (point clouds, surface normals, depth images, curvature images, and 3D voxels) have similar identification performances provided that its matching feature is selected and that the gallery contains at least two data items per subject. Instances of a matching feature are the following: DCT or DFT features for depth images, shape

index for curvature representation, and NMF for point cloud.

The fusion of intelligently selected experts improves the recognition performance, where additional 7.14 points of accuracy is gained for FRGC v2.0. Inviting everybody is not necessarily a good idea, an expert-selection algorithm, such as Sequential Floating Backward Search, works better.

5. REGION-BASED RECOGNITION OF 3D FACES WITH EXPRESSION VARIATIONS

In this chapter, we propose the application of masks as a means to mitigate expression-distortions on 3D faces and to enhance their recognition performance. Masking becomes necessary to de-emphasize the face regions that deform under expression. We have conducted experiments with various masks, namely, ellipse-shaped binary masks, Gaussian, super- Gaussian and raised-cosine masks. The design issues of the masks, such as the mask size, the centre, the support region, the decay rate of the tails, etc. are studied and adjusted with respect to their recognition performances. We show first that warping the depth values of corresponding face points onto the same spatial coordinates while obtaining the 2D depth images is beneficial, and second, that proper masking can add several percentage points to the recognition performance.

5.1. 2D Depth Image Generation

The common approach for registration is alignment of the 3D point cloud of a probe image onto each gallery image separately via the Iterative Closest Point (ICP) algorithm [95]. Since ICP is a time-consuming procedure, the alignment of an input face to all the faces in the database precludes real-time operation. Therefore we use an Average Face Model (AFM) obtained from a set of training face samples and align the 3D point cloud of each face only to AFM via ICP. Figure 5.1 shows an Average Face Model mapped onto a 2D depth image. This scheme allows us to rapidly build correspondences among faces.

ICP alignment is a rigid transformation that yields aligned point set correspondence of a face. We first use the ICP algorithm to best match the fiducial points of a given face to those of the AFM. The seven fiducial points used are the four inner- and outer-eye corners, nose tip and the two mouth corners. Then we apply spatial warping to relocate (x, y) face coordinates on top of the regular grid of the AFM.

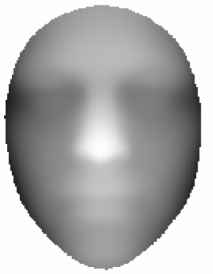


Figure 5.1. Depth view of the AFM.

Finally, the registered depth image of a face is formed with the z-coordinates of the input face image located at the (x, y) coordinates of the Average Face Model to yield the depth function $H(x, y)$. This idea is similar to the Active Appearance Model of Cootes et al. [14], where 2D intensity faces are warped on an average shape model of the faces in order to establish correspondences. In this thesis, we treat the depth of each point as the appearance of a face. The model will be complete if we also consider the (x, y) coordinates of the face points and model the spatial arrangement of the points. However in this thesis, we limit ourselves to the depth values only. Figure 5.3 shows the warped depth images of the faces depicted in Figure 5.2. The faces look very similar to each other, because the spatial arrangements of the pixels belong to the average face. However, the geometric information represented by the depth values is preserved. Figure 5.4 shows the profiles of three face images of a subject in dashed curves and three profiles of another subject in solid and black curves. With this single profile, two classes seem to be separable from each other.

This warping scheme not only moves corresponding face points to the same spatial locations in the depth image, but also reduces the deformation caused by expression variation. A visual inspection of Figures 5.2 and 5.3 shows that the within-class variations due to expression are reduced after warping. This result is coherent with the Active Appearance Model of faces [14], where by warping intensity values on to an average shape model, one can decouple expression from the appearance of the face.

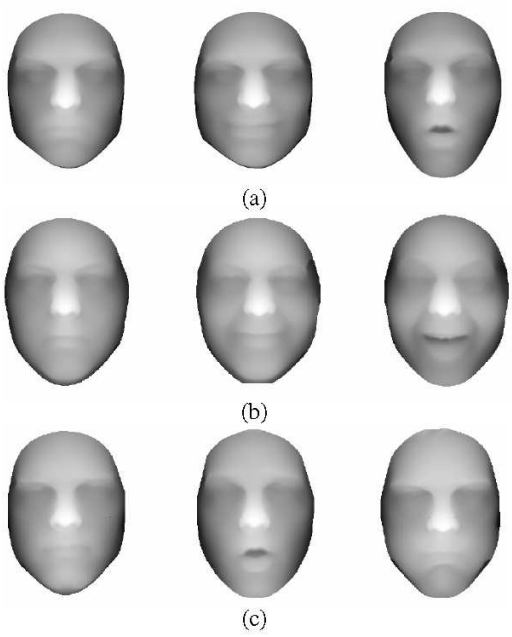


Figure 5.2. 3D faces from three different subjects (a, b, c). Faces on the same row correspond to the same person with different facial expressions.

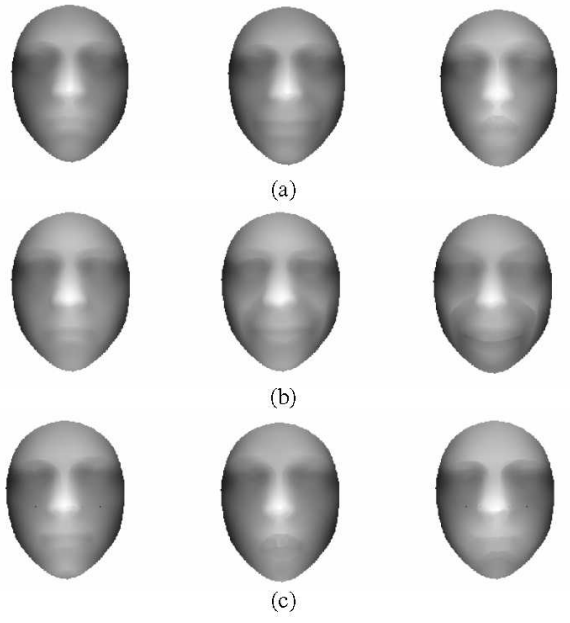


Figure 5.3. Warped depth images of the faces shown in Figure 5.2.

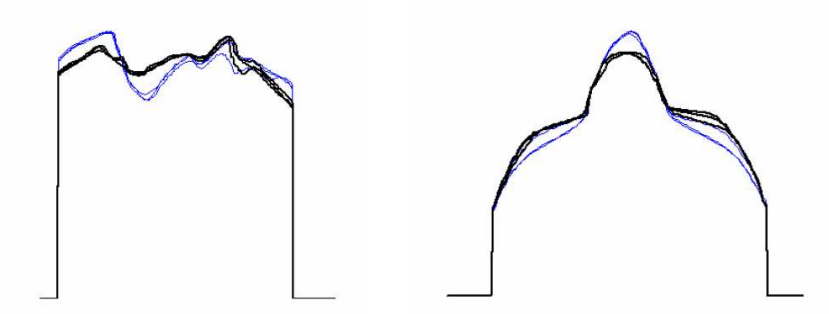


Figure 5.4. Vertical (a) and horizontal (b) profiles of faces from two subjects.

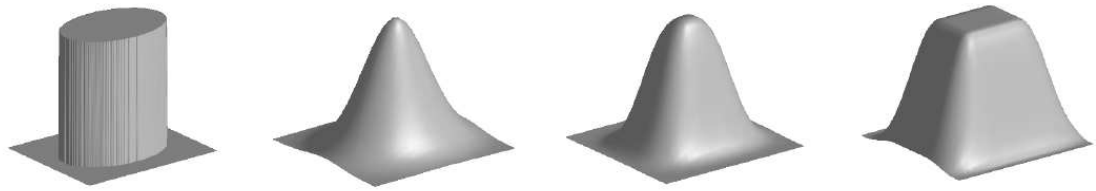


Figure 5.5. Ellipse-shaped (a), Gaussian (b), super-Gaussian (c) and raised-cosine (d) masks.

5.2. Masking Schemes

Since regions of the depth map $H(x, y)$ have varying reliability, we can privilege certain regions over others by multiplying with masks $W(x, y)$:

$$I(x, y) = W(x, y)H(x, y) \tag{5.1}$$

The two issues that must be addressed are the shape and the location of the mask functions. We have tested four different masks: Ellipse-shaped binary mask, Gaussian mask, super-Gaussian mask and raised cosine mask (Figure 5.5).

The ellipse-shaped binary masking can be considered as a parts-based approach, where one particular region of the face is matched with the corresponding region of another face. We have chosen ellipse-shaped regions in order to make a fair

comparison with the Gaussian and super-Gaussian counterparts based on similar control parameters, such as centre, size, support region, etc. The general form of the ellipse- shaped binary mask is as follows:

$$W_E(x,y) = \begin{cases} 1 & \text{if } \left(\frac{x-X_c}{a}\right)^2 + \left(\frac{y-Y_c}{b}\right)^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

We have selected three parameters of the ellipse as variables: The vertical centre point of the ellipse, Y_c , along the symmetry axis of the face, the horizontal radius, a and the vertical radius, b . The centre of the ellipse is constrained to be at the symmetry axis of the face. The Gaussian mask has the following form:

$$W_G(x,y) = \exp\left\{-\left(\frac{x-X_c}{a}\right)^2 - \left(\frac{y-Y_c}{b}\right)^2\right\} \tag{5.3}$$

The Gaussian mask is applied to the whole face; hence this scheme does not discard any face region. Instead, we weight the face points smoothly, with the points near the centre of the mask contributing more as compared to further points. This is controlled by the aperture parameters of the Gaussian mask.

To manipulate the decay regime of the Gaussian mask, so that it remains flat over a larger region and then drops more rapidly to zero, we propose the use of a super- Gaussian mask of order three. Higher powers of the super- Gaussian will make the mask similar to an ellipse-shaped mask.

$$W_{SG}(x,y) = \exp\left\{-\left|\frac{x-X_c}{a}\right|^3 - \left|\frac{y-Y_c}{b}\right|^3\right\} \tag{5.4}$$

The fourth type of mask is the raised-cosine mask, which can provide a flat value over a controlled support region. The raised-cosine mask can be obtained from the multiplication of raised-cosine windows along rows and columns of the

image:

$$W_{RC}(x, y) = W^a_{RC}(x) W^b_{RC}(y) \tag{5.5}$$

where,

$$W^K_{RC}(t) = \begin{cases} 1 & \text{if } |t| \leq \frac{K(1-\beta)}{2} \\ g^K_{RC}(t) & \text{if } \frac{K(1-\beta)}{2} < |t| \leq \frac{K(1+\beta)}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

and

$$g^K_{RC}(t) = \frac{1}{2}\left[1 + \cos\left(\frac{\pi K}{\beta}\left[|t| - \frac{1-\beta}{2K}\right]\right)\right] \tag{5.7}$$

We have set β to 0.5. The raised-cosine mask provides a region-based representation similar to the ellipse-shaped binary mask. However, with the raised-cosine mask, we have a smoother transition between the support region and other regions of the face.

5.3. Features

We have tested the performance of masking schemes with DFT and PCA. We apply 2D-DFT on the registered and masked depth function and extract the first $M \times M$ complex DFT coefficients. The real and imaginary parts of these coefficients are concatenated in a one-dimensional vector, which forms the DFT-based feature vector of a masked face. For PCA, the values of each of the masked faces in the training set are concatenated to form a single vector. Part of these vectors are used as training vectors to constitute the PCA bases, while the remaining ones are projected onto these bases to form the feature vectors of the test faces. Furthermore, the DCT and PCA coefficients are reweighted through QR-decomposition in order to make use of the class information available in the training set.

Table 5.1. Recognition performances of unmasked faces on the FRGC v2.0.

	DFT	PCA
Unwarped	71.71 %	74.20 %
Warped	80.66 %	87.15 %

5.4. Experimental Results

We have tested the performance of masking-based 3D face recognition on the FRGC v2.0 database. We have considered the case where there is only one gallery image in the database. There are 410 subjects hence, 410 gallery images. The remaining 3542 face scans are used as test images. In order to train the PCA basis and obtain QR decomposition we have used a separate dataset: The FRGC v1.0 database. This database consists of 854 face scans of 194 subjects and does not contain the face scans present in FRGC v2.0. The PCA basis and the transformation matrix R are calculated and fixed on the v1.0 database, and then used to weight the features of the gallery and test images of the v2.0 database. As a baseline, we have used both warped and unwarped depth images without masking. Table 5.1 shows the performances obtained with unmasked face images using DFT and PCA-based features. The best performance on unmasked images is obtained with warping and PCA-coefficients. This is much higher than the best performance obtained from the DFT coefficients with masking (Figure 5.6 and Table 5.2). This is not surprising, since the 2D-DFT is sensitive to spatial structure of the depth values, whereas PCA only considers the variations among corresponding points regardless of their position. After warping, the spatial structure of the depth values does not carry class information since they are arranged with respect to the average face.

By varying the vertical centre, the support regions and the decay rates, we have experimented with 128 variations of each of the four masks. The depth image is of size 201×161. We varied the centres of the masks between 30 and 160, with an increment of 10. The a and b parameters for the elliptic, Gaussian and super-Gaussian windows are taken in the range of 20 to 80 with an increment of 20. For the raised-

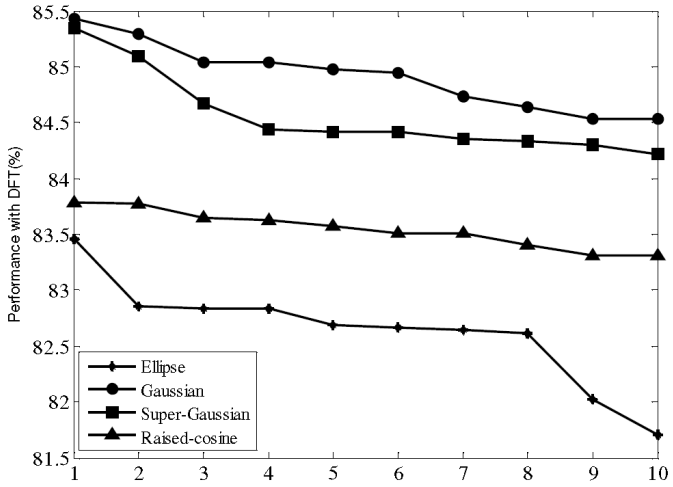


Figure 5.6. Performances with best 10 mask parameter sets for each masking scheme, obtained with DFT coefficients.

Table 5.2. Recognition performances of best masks with DFT in percentages.

Method	Unmasked	Ellipse-shaped	Gaussian	Super-Gaussian	Raised-cosine
DFT	80.66	83.46	85.43	85.35	83.79
PCA	87.15	87.32	88.09	87.89	87.63

cosine mask, a and b vary between 60 to 240 with an increment of 60. Table 5.2 gives the best performances of the four masks among their different parameterizations with DFT and PCA features. For DFT, unmasked image performance is 80.66 per cent, and all masked versions register a few percentage point improvement. The Gaussian mask has the highest gain, followed closely by super-Gaussian. Both raised-cosine and elliptic windows fall about two percentage points behind. Both Gaussian and raised-cosine masks are quite insensitive to parameter adjustments while the elliptic mask necessitates fine-tuning (Figure 5.6).

Compared to DFT features, the gains with the PCA features are less impressive. The best performance is again achieved with the Gaussian mask. Elliptic binary masking yields little improvement. Figure 5.7a and Figure 5.7b illustrate the best

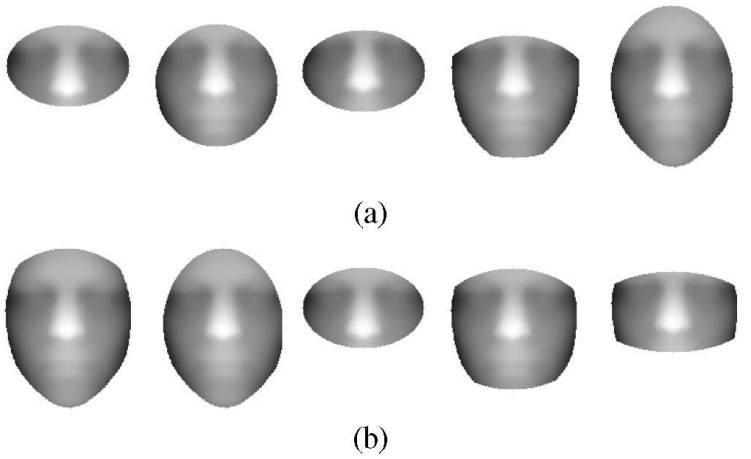


Figure 5.7. Ellipse-masked faces giving the best five performances with (a) DFT coefficients (b) PCA coefficients.

five ellipse-masked faces with for DFT and PCA techniques. For the DFT technique, the best ellipse includes only the nose and eye regions. The second runner ellipse includes also the mouth. This result shows that discarding the mouth and chin for the sake of expression invariance causes a loss in the class information available to a recognition system. Actually, when we observe the best ellipse-masked face for the PCA case (Figure 5.7b, first face) we see that almost all face regions contribute to this performance.

The PCA coefficients derived from the Gaussian masked faces give the best performance, and the performance is relatively insensitive to the parameterization. Figure 5.8 shows the Gaussian-masked faces giving the best five performances, all of which are around 88 per cent. Their centres are all located around the nose tip. However their aperture parameters are different.

5.5. Conclusions

In this chapter, we have proposed the use of smooth masks to deal with expression variations in 3D faces. We have conducted experiments with an ellipse-shaped binary mask, a Gaussian mask, a super-Gaussian mask and a raised-cosine mask

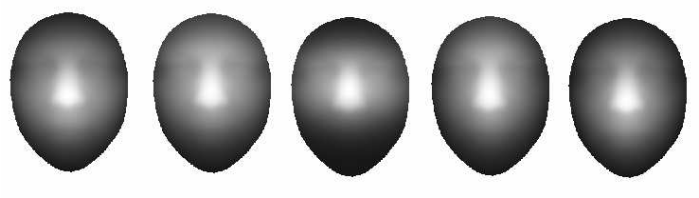


Figure 5.8. Gaussian-masked faces giving the best five performances with PCA coefficients.

with a large number of possible parameters for each. We have also experimented on the use of warping depth fields into an average face in order to reduce the deformation due to facial expressions.

Warping depth images so that the depth values at the same location come from the corresponding points of the 3D point clouds is beneficial for reducing the effect of facial expression. This scheme is of great advantage especially with the PCA-based technique, since PCA models the data better when correspondences are well-established.

Another important observation is that, avoiding expression- prone face regions such as mouth and chin, results in class information loss. Weighting the face regions smoothly via a Gaussian mask, with high weights assigned to the rigid regions such as nose tip, results in an improved performance. Furthermore one does not need to fine-tune the parameters of the Gaussian mask in order to get the best region. The best performance in the literature with the same database and the same experimental setting is 94.9 per cent [106]. Here, various face regions are compared with each of the gallery images via ICP and the results are fused with committee voting. We have obtained 88.09 per cent recognition performance with warping and Gaussian masking. We have implemented only one ICP procedure for a probe image, which makes the system much faster and we have used a single masked image. The proposed schemes are open to improvements.

6. INDEXING AND RETRIEVAL OF 3D MODELS

6.1. Introduction

The technological advances enabling fast and reliable 3D acquisition and reconstruction of objects have resulted in rapidly growing databases of 3D models in many domains. This increase brings up a need for efficient tools for indexing the objects for various recognition, classification or retrieval tasks. Manual annotation of objects with keywords has several limitations, such as being labor intensive and difficult to keep updated, the dependency of the choice of keywords to one particular application, and the insufficiency of keywords to describe the shape of an object. These limitations have made automatic indexing and retrieval of 3D objects a popular current research topic.

Automatic and fast retrieval of three-dimensional objects from large databases is becoming more vital with the increasing number and scope of 3D object models in computer applications such as CAD/CAM, 3D games, virtual reality media, biomedicine, and virtual museums. Therefore, it is necessary to build efficient indexing schemes that exploit discriminatory shape characteristics of different object categories.

The retrieval of general 3D models is a hard problem since its evaluation is highly objective, in the sense that the categorization of the models and the similarity judgements depend on the user's expectations. This is in contrast to the case with hand-based or 3D face-based biometry, since the similarity notion between two samples is well-defined: "They belong to the same person or not". On the other hand, the retrieval of generic 3D models has the same challenge as the image retrieval problem has: The semantic gap, which is defined as "the difference between the contextual and semantic knowledge of an entity described in natural language and its computational representation in a formal language" [123]. The formal language in 3D object retrieval may correspond to the procedures or rules that produce a

set of measurements and their transformations, graphs, look-up tables, similarity scores, search strategies and so on. The contextual and semantic knowledge is any word, phrase or sentence in natural language pointing to or related to an object. Examples of such expressions are ”a cat”, ”a tabby cat”, ”a fat tabby cat with large ears, sitting on a chair”. Despite the challenge, computational tools or descriptors are being developed to map the geometry of objects to similarity scores among them and rigorous semantic categories are being defined in order to evaluate the success of these mappings.

In this chapter, we explore subspace approaches for the shape-based classification and retrieval of complete 3D object models. This approach is based on the conjecture that 3D shapes are compressible or redundant in that they can be characterized with fewer coefficients as compared to their voxel data size. We demonstrate the potential advantages of data-driven methods that better capture the statistical characteristics of the 3D objects in retrieval tasks. Among many possible subspace techniques in the literature, we concentrate on three widely used and well-studied subspace methods, namely, PCA, ICA, and NMF, which we have introduced in Chapter 2.

Subspace-based techniques have commonly been used for characterization of 3D anatomical structures in biomedicine, head and body recognition applications [124, 125, 126, 127]. However, to the best of our knowledge they have not been extensively studied specifically for indexing and retrieval of general 3D models. These methods are generally considered as domain dependent; they may not generalize for totally different structures that were not represented in the training set. For example, a subspace that is built using only human models cannot generalize for totally different structures, such as plants. With this proviso, subspace methods are otherwise powerful in characterizing the statistics of the data and in retrieval if the test models are well represented in the training set. They greatly reduce the dimensionality of the models supplying compact representations, which enable fast searching. The feature extraction procedure is also time efficient since it only involves multiplication with a matrix.

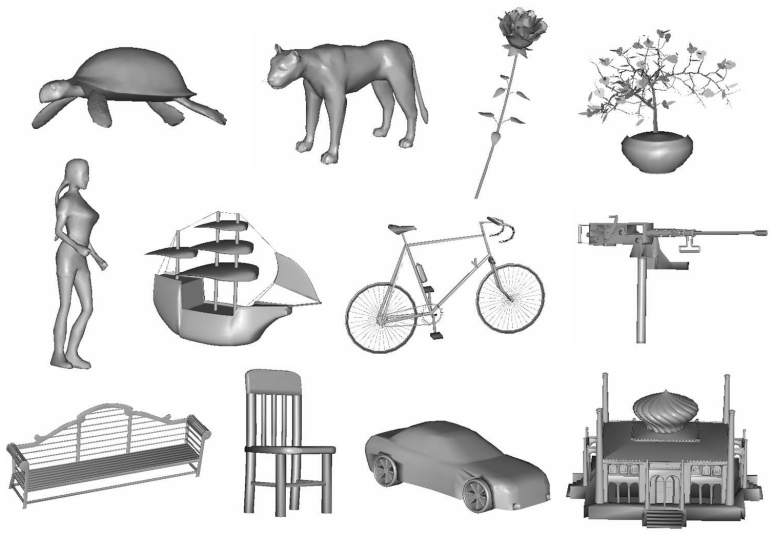


Figure 6.1. Samples of general 3D models.

The subspace-based retrieval methods that we propose are not rotation invariant, hence their success depends critically upon the quality of the object pose normalization or registration. Pose normalization is crucial, since misalignments lead to eclipsing of genuine shape variation by pose variations, that is, translation, scaling and rotation. In specific categories of shapes, such as 3D faces, body shapes or organs, alignment and correspondence building are performed with the aid of anatomical landmarks. However, general object classes, such as buildings, plants, and animals lack such common natural landmarks For mixtures of object classes, even if landmarks exist for some classes, they are not easily generalizable to other classes (Figure 6.1).

We address the alignment problem with the aid of Continuous PCA (CPCA) [128]. Furthermore, the distance transform of the voxelized models provides a smooth function with an inherent robustness against minor misalignments. Finally, we resolve the ambiguity in pose by exhaustively searching over all possible mirror reflections and axis re-labelings as shown in Section 6.6.

We introduce a general subspace-based framework [129] for indexing of general 3D objects. We propose and explore various alternatives for each component of

this framework, namely for data representation, object alignment, choice of the subspace and shape matching. In particular, we use the inverse of the distance transform for data representation, which offers a good compromise between fast decay rate and large support (Section 6.3.3). For alignment of training models, we propose mean shape-based and class-based correction schemes to resolve pose ambiguities resulting from CPCA (Section 6.5.4). For shape matching, we propose a computationally efficient version of the Munkres algorithm, which we refer to as the pose assignment strategy, in order to compute the distance between two models by taking into account all possible mirror reflections and axis re-labelings (Section 6.6). As a result, the PCA, ICA and NMF subspaces, when tailored to the needs of a retrieval problem and applied to voxelized 3D shapes (Section 6.5), provide state-of-the-art performance. The retrieval performance of the proposed framework is demonstrated on Princeton Shape Benchmark (PSB) database. The subspace-based methods, when combined with other state-of-the-art descriptors in the literature, achieve the highest retrieval performance reported so far on PSB test set.

6.2. Related Work

The last decade has witnessed the emergence of a new research area in computer vision, the query by content of general 3D models from large databases, with the introduction of the Nefertiti project by Paquet et al. [130]. The large amount of research carried on within the last ten years is thoroughly categorized and reviewed in a number of survey papers [131, 132, 133], and PhD theses [128, 134, 135, 136].

In this brief survey, we focus on the problem of retrieval of objects belonging to general categories, such as cats, tables, airplanes, etc. This type of categorization is subjectively plausible in that it corresponds to what we would picture in our mind while searching an object in the Web. A significant effort has been dedicated in the literature to obtain rotation-invariant features from 3D objects. For example, Zaharia and Preteux [137] use shape index histograms to compare the objects. While the shape index, based on principal curvatures, is a powerful object surface attribute, it is computationally tedious and also quite sensitive to noise and resolution level.

Osada et al. [138] use various shape functions, such as distance between two arbitrary points on the object surface. The sample distribution of these shape functions then become object signatures. This distribution-based approach is appropriate for shape categorization, for example, used as a pre-classifier, but not for object identification.

Converting the surface representation of an object, mostly from a mesh representation into a voxel grid has been suggested by many authors [128, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148]. We also base our data structure on a regular voxel grid since it provides a parametric representation of the surface as a three-dimensional function that is well-suited for subspace analysis. Furthermore, voxelization yields uniform sampling of the object surface that is originally defined by the non-homogeneous, disoriented and topologically inconsistent mesh models referred to as "polygon soups" [142].

Funkhouser et al. [142] suggested the use of a binary voxel grid, where the voxels that intersect the object surface are assigned the value one. Vranic [128] has argued that a binary function will result in a loss of important surface information and proposed to use a real function, where each voxel is attributed to a value proportional to the area of the surface patch confined in it. Novotni and Klein [145] suggested to voxelize the surface using radial linear, binary and Gaussian kernels; however they obtained the best results with the binary kernel. Kazhdan et al. [149] proposed the exponentially decaying Euclidean distance transform. Although the authors used this method in order to enable the computation of their reflective symmetry descriptor for models with irregular meshes (topologically inconsistent, with cracks and flipped triangles), the distance transform has many other advantages as pointed out in [128]. The use of distance transform instead of plain surface voxels alleviates the negative impact of pose variations on shape matching. Such variations, though can be minimized via pose normalization techniques, are usually inevitable in retrieval systems. The use of an adequate distance transform is even more crucial in our case since it is a well known fact that subspace-based methods are usually very sensitive to pose variations. We evaluate experimentally the retrieval

performances of various 3D functions based on the distance transform on the training set of Princeton Shape Benchmark. Since the inverse of the distance transform provides the best results, we use it as the input data to our subspace analysis.

One source of controversy in the 3D model retrieval community concerns the pose invariance problem. Some authors advocate the development of pose-invariant descriptors [142, 145, 149, 150] while others rely on preprocessing for pose normalization and then extract pose-dependent features from the normalized representations [128, 151, 152, 153, 154]. Since our subspace-based features are dependent on the pose, we correct the pose of the model prior to voxelization.

Pose normalization techniques can be listed as PCA, weighted PCA [151, 155], Continuous PCA [128, 156] and PCA on the normals of the model (NPCA) [154]. All these techniques aim to transform objects into a canonical coordinate frame so that the normalization of each model becomes totally independent from other models. Among these, CPCA is the most robust method, since it incorporates the whole object surface to the pose normalization via integration over triangles, instead of just using the triangle vertices.

An alternative to rotation-invariant features is to obviate the rotation uncertainty. Thus an object can be aligned along its principal axes, e.g., its principal components. Paquet et al. [151] construct three cords-based histograms after PCA-based alignment. Ricard et al. [157] utilize magnitudes of 3D ART coefficients applied to the voxelized objects as object descriptors. Since magnitudes of 3D ART coefficients are only invariant to rotations around z-axis, these authors align the objects principal axis with the z-axis prior to computation of ART coefficients. In the same vein, Vranic and Saupe [158] take the 3D-DFT of the binary voxel representations. Since the 3D-DFT coefficients are not rotation invariant, DFT is applied after alignment to principal axes. Vranic and Saupe [156] have also experimented spherical harmonics expansion with the PCA aligned objects.

The descriptors that are closely related to our subspace-based approach are the

representations of the 3D models in some transform domain. In general, transform-based methods assume the following signal model:

$$\mathbf{x} = \phi\mathbf{b} + \mathbf{N} \tag{6.1}$$

where, $\mathbf{x}$ is the data representing the geometry of a model, ϕ is the set of basis vectors onto which $\mathbf{x}$ will be projected; and $\mathbf{b}$ is the coefficient vector and finally $\mathbf{N}$ is the observation noise. The aim is to describe the shape in a compact form that preferably possesses an inherent multiresolution nature. Spherical harmonics-based analysis has been used as shape descriptors in many works [128, 142, 156, 159, 160, 145, 161]. Vranic [128] suggested the 3D - DFT to characterize the 3D voxel grid. Novotni and Klein used Zernike functions, which are basically spherical harmonics modulated by appropriate radial functions [145]. Ricard et al. introduced 3D Angular Radial Transform, which is defined as a product of radial and angular basis functions [143, 157].

In any such transform-based representation, the discriminating shape information is subsumed in the coefficients $\mathbf{b}$ while ϕ is fixed. Most of these transform domains are constructed in terms of complex exponentials and sinusoids of varying frequency. One of the drawbacks of using harmonics as basis functions is that it is difficult to obtain a compact representation of a 3D shape with high frequency content. If the surface is composed of a series of jagged or highly curved concave and convex parts, as in articulated objects, many coefficients are required to describe the model. An additional drawback of the spherical harmonics descriptor is the necessity to describe the geometry of the object in terms of functions on a sphere. However, most of the 3D models cannot be mapped onto a single sphere without loss of information. The common approach is to construct concentric spheres of various radii, centered at the center of mass of the object, and define separate spherical functions using some projection of the object geometry onto the spheres [128, 142]. Each sphere is encoded with spherical harmonics independently from others. This procedure brings an artificial partitioning of the model and is sensitive to parameters such as the scale of the model, its center of gravity and the number of the spheres.

Since PCA alignment may give nonconsistent orientations within a class [142], there have been attempts to extract rotation-invariant features from transform coefficients. Novotni and Klein [145] use 3D Zernike moments as descriptors for 3D shape retrieval. Kazhdan et al. [161] derive rotation-invariant features from spherical harmonic coefficients. They first rasterize objects in a voxel grid to obtain a 3D binary function, and then compute spherical harmonic invariants of the binary on concentric shells.

On the other hand, subspace techniques such as PCA, ICA and NMF are data-driven approaches and solve jointly for the subspace spanning vectors and their projection coefficients. These techniques exploit the second or higher order statistics of the data to extract the subspace information, that is, the basis functions. So far they have been used only to model 3D objects of the same genre. For example, there is a vast literature of subspace analysis of anatomical structures in the domain of biomedical imaging [124, 125, 126]. These techniques usually concentrate on a single structure, such as the corpus callosum, and model the small, but medically significant variations. Likewise, in biometric systems that identify a person from her 3D face geometry, subspace methods are powerful tools for modeling the interpersonal variations [162, 92, 127]. There is also research for modeling shape variations of 3D human body via PCA [163] and human torso via PCA and ICA [164]. However, to the best of our knowledge subspace methods have not previously been considered for describing general 3D shapes.

6.3. Voxel Representation

6.3.1. Pose Normalization

In order to normalize the triangular mesh models before voxelization, we use the Continuous Principal Component Analysis (CPCA) technique developed by Vranic et al. [128, 156]. The aim of this procedure is to transform the mesh model into a canonical coordinate frame. The model is first translated such that its center of gravity coincides with the origin. Scale invariance is achieved by setting the area-

weighted radial distance from the origin to unity. Then, the covariance matrix of the x, y and z coordinates on the object surface is estimated via a continuous integration over all the triangles. The eigenvectors of the covariance matrix are considered as the principal axes, and the object is rotated such that the canonical coordinates coincide with the eigenvectors.

The eigenvectors of the covariance matrix are sorted in decreasing order of their eigenvalues, and they are made to correspond to the x, y and z axes of the canonical frame, respectively. This procedure assigns the orientation of the largest spread of the surface points with the x-axis, the next largest spread with y axis and so on. After the order of the axes is determined, the second order moments of the model are used for selecting the positive direction of the axes [128].

In [142], the authors point out some problems associated with PCA-based pose normalization techniques. For example, the eigenvalues may be multiple, or too close to each other for models with high symmetry, such as a cube or a right square prism. Isotropic models, i.e., models that are close to a sphere, may not even possess strong principal orientations. Another more serious drawback of PCA normalization is its potential risk to put objects of the same class "out of phase" due to inconsistent axes labelings and reflections [152].

Nevertheless, CPCA is a practical and powerful technique for pose normalization. It gives small alignment errors, especially for objects that have clear principal directions [165], and its merit has been proved by the high retrieval performances achieved using pose-dependent shape descriptors [128, 152, 153, 154].

6.3.2. Binary Function in 3D Space

The voxelization of a triangular mesh model is a re-sampling process. Regardless of the degree of its irregularity, the mesh model can be seen as a piecewise continuous surface in 3D space. The voxelization converts the mesh information into a discrete function regularly sampled in 3D coordinates. The mesh model is

placed in a Cartesian grid at some resolution and voxels are assigned the value one if the surface passes through it, and zero otherwise. The resulting discrete function is a 3D binary function, which is a distorted approximation of the object. Some information is lost during sampling, and artifacts, such as aliasing, are introduced due to the coarse structure of voxel grids.

The voxelization operation involves setting of two important parameters: The first one is the size of the rectangular prism, a 3D windowing function, in which the object will be sampled. The second one is the sampling density, the number of voxel units, along each direction.

Using the bounding-box of the object itself as the windowing function, will make the representation sensitive to outliers. Since the number of voxels must be the same for all the objects, fitting an object into its bounding box will scale it with respect to its extremities. Instead, we scale objects such that their area-weighted mean distance (AWMD) from the center of gravity to the surface is set to unity. Then we put the object in a fixed size box and discard all object parts that fall outside the box Figure 6.2.

We use a cube as the box; hence we take identical dimensional factors along x, y and z directions. We define the size of the box as half the length of one of its edges. There are obvious trade-offs in the choice of the box size vis-á-vis the normalized scale. The choice of a large box means larger voxels and, a coarser representation; on the other hand, small boxes may crop some important model parts. In extremum, the cube size can be adjusted to encompass all extremities of all the objects in a database. Then we guarantee to have all objects remain within the box, while sacrificing resolution, and for most objects in the database leaving unnecessarily significant parts of the box volume empty.

In this work instead, we search for optimum box size that includes a proportion of objects within the box favorable to good classification. We select the box size as some factor of AWMD. Since AWMD is already set to unity for all the models during

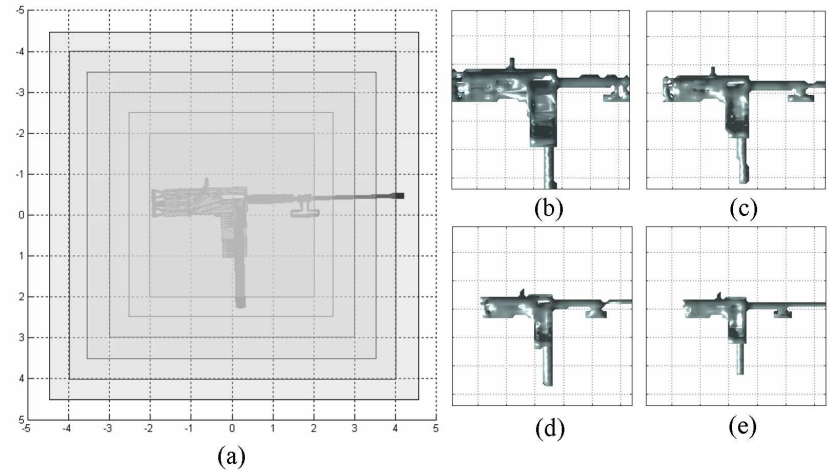


Figure 6.2. Selection of the box size for voxelization of the mesh model in (a). The resulting voxel representations are shown on the right, with box sizes of 2.0 (b), 2.5 (c), 3.0 (d) and 3.5 (e).

scale normalization, the size of the box is equal to the factor we choose (Figure 6.2). We have used two approaches to set the box size. The first approach examines the histograms of extremities along x, y and z axes of the pose normalized objects in the database and selects the box size such that the majority of the objects will remain entirely in the box. The second approach, given a fixed-size box, calculates the cropped proportion of objects in terms of surface area and then chooses a box size to keep the lost surface ratio below a threshold. We have applied both procedures to the training set of Princeton Shape Benchmark and have chosen the latter method since it is more robust to outliers. Details are given in Section 6.7.1.

We rasterize scale-normalized objects into grids of $R \times R \times R$ voxels. The number of voxels determines the level of detail that will be preserved in the voxel representation. While in computer graphics both high resolution and anti-aliasing filtering are required to obtain visual quality, for classification and retrieval purposes, a rough approximation may suffice depending on the application. The sampling density is a compromise between maintaining class-specific details and glossing over small within-class variations, which are considered as ”noise”. Too small an

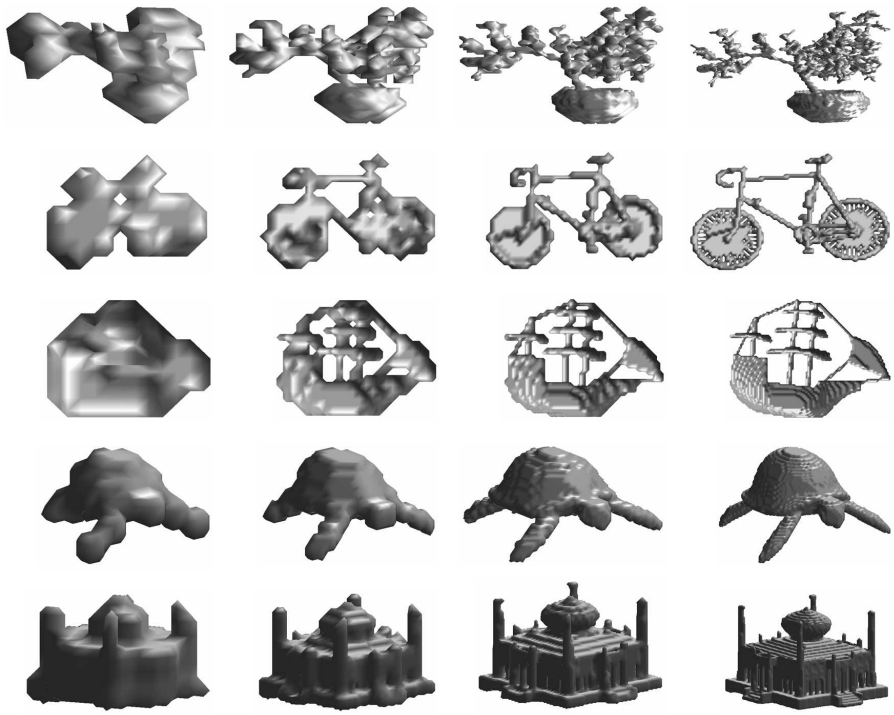


Figure 6.3. Models voxelized at resolutions R = 16, 32, 64 and 128 from left to right.

R obviously results in a rough voxelization; on the other hand, too large R values, while attaining fine voxelization, may unnecessarily bring forth disparities due to slight pose normalization errors. This issue of mismatching of two similar high resolution models is discussed in [128]. Vranic has suggested starting with high resolution volumetric representations and to suppress noise and uncertainty during feature extraction, e.g. filtering of 3D-DFT coefficients of the volume [128]. However, high-resolution volumetric models demand more storage and processing time, both during preprocessing and feature extraction stages.

Figure 6.3 shows voxelized representations of five models with various selections of R. For these specific examples, representations at resolutions of R = 32 or 64 seem to be sufficient to at least visually identify object classes. In our work, we conducted experiments with R = 32.

6.3.3. Functions of the Distance Transform

We found it useful to propagate the binary shape information via 3D distance transform to the 3D space where the object is rasterized. Binary voxel representation will result in most of the voxels to have zero value and these voxels will not carry any information about the structure of the object. One consequence of binary representation is that it is not sufficiently robust against pose perturbations. The distance transform has many advantages over the binary function. First, the representation is smoothed and high-frequency artifacts due to the blocky structure of the binary voxels are suppressed. Thus contradictory indications by the nearby binary voxels of two objects, an artifact of binary voxelization, will be avoided. Second, each voxel in the cube will contribute to the distance computation between two objects.

The distance transform, also known as the distance field, is a function which maps each point in the space to the distance between that point and the nearest non-zero point in the original binary function. We can define the 3D distance transform, $DT_f(p)$ at point $p = (x, y, z)$ of the binary function $f(p)$ as

$$DT_f(p) = \min_{\{\hat{p},\, f(\hat{p})=1\}} d(p, \hat{p}) \tag{6.2}$$

For distance measure $d(p, \hat{p})$ we use the Euclidean distance. The distance transform is zero at the surface of the object and increases monotonically as we move further from the surface. The values can become quite large at the borders of the box. Thus points farthest from object surface will have higher impact on the shape comparison, which is counterintuitive. We prefer, therefore to use a function of the distance transform that takes its largest value on the surface of the object and decreases smoothly as one moves away from the surface. We have experimented with the distance transform itself and the following functions of it:

- The inverse of the distance transform (IDT):

$$IDT_f(p) = \frac{1}{1 + DT_f(p)} \tag{6.3}$$

- The Gaussian of the distance transform (GDT):

$$GDT_f(p) = \exp\{-\left(DT_f(p)/\sigma\right)^2\} \tag{6.4}$$

  where the parameter σ determines the width of the Gaussian profile.

- A piecewise linear function of the distance transform (LDT):

$$LDT_f(p) = \begin{cases} 1 - \frac{DT_f(p)}{k} & \text{if } DT_f(p) \leq k \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

  where the parameter k determines the width of the triangular profile of the linear function.

Kazhdan et al. [149] used an exponentially decaying function of the distance transform, which corresponds to our GDT. However, they fixed the width of the Gaussian with respect to the average radial distance of the object. We have performed experiments with various radii of the Gaussian function. Our results show that the inverse of the distance transform gives significantly better results regardless of the resolution of voxel representation.

Figure 6.4 shows the profiles of the functions with various width parameters, σ and k. The Gaussian and linear profiles are similar in their appearance, and in fact they yield similar retrieval performances (Section 6.7.2). If their support is small, they decay rapidly toward zero. For larger widths, the GDT varies slowly in the neighborhood of the surface, which in turn causes blurring of the object surfaces (Figure 6.5h). The profile of the IDT is significantly different from the others. First, it decays rapidly in the beginning, thus the blurring effect is mitigated; furthermore voxels on the surface gain much more. Thus the relative importance of the surface

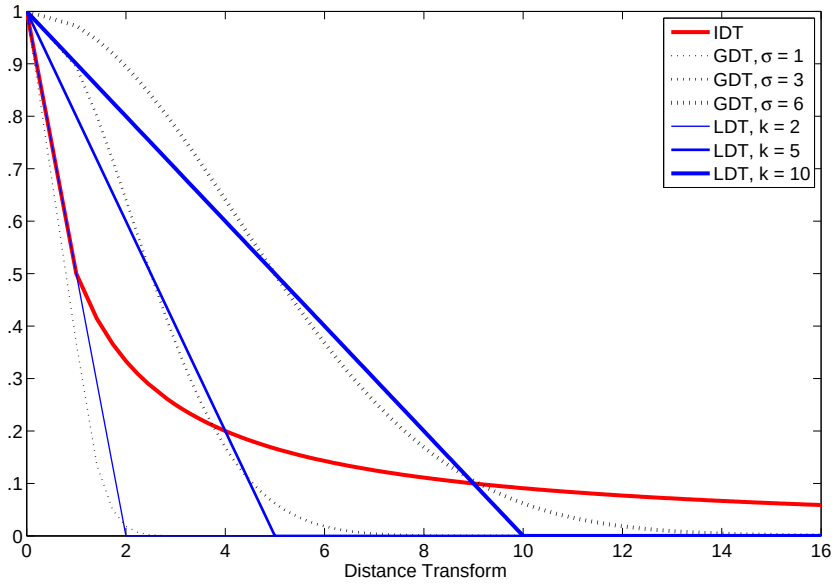


Figure 6.4. Profiles of the functions of distance transform with various parameters.

voxels of the object is preserved. Second, IDT has a larger effective support than GDT and LDT. Therefore the distance information is propagated further away from the object surface, but with attenuated weights as compared voxels proximal to the surface.

Figure 6.5 shows the voxel representation of a chair and the slices from various possible 3D functions. The slice from the binary representation carries very little information about the general shape of the model. GDT and LDT functions with small support are similar to the binary function, since only very prominent voxels to the surface are weighted. Farther away voxel attributes drop to zero. On the other hand, increasing the support of GDT and LDT causes a blur of the representation. Hence IDT is a good compromise between fast decay rate and large support.

6.4. Direct Voxel Comparisons

Direct comparison of objects provides a base-line to measure gains enabled by the feature extraction schemes. The representation modalities can be voxel-wise

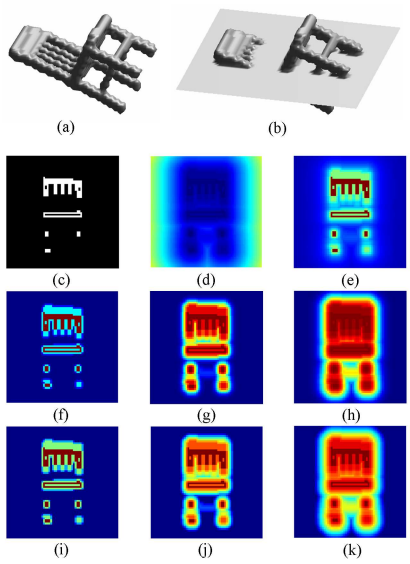


Figure 6.5. Slices for the chair model (a) extracted as in (b). Slices from binary function (c), distance transform (d), Inverse Distance Transform (e), Gaussian of the distance transform with $\sigma = 1$ (f), $\sigma = 2$ (g), $\sigma = 6$ (h), piecewise linear function of the distance transform with $k = 2$ (i), $k = 3$ (j), and $k = 10$ (k).

differences of volumetric models or pixel differences of depth buffers, etc. This gain is expressed in terms of increased discrimination power and decreased search effort. All feature extraction or selection methods focus on class-specific shape characteristics and attenuate irrelevant variations and details. Subspace projection as a feature extraction method provides a controlled way of filtering details non pertinent to classification. In order to measure the performance gain, if any, of the subspace algorithms, we resort to baseline retrieval performance obtained directly via raw data without any feature extraction attempt.

For the RxRxR voxel array representation, we convert this 3D array to a 1D vector, $\mathbf{x}$, using lexicographical ordering with indexing m. The distance of a query model $\mathbf{x}_i$ to a target model $\mathbf{x}_j$ in the database is the sum of pairwise absolute differences of the attributes of the voxels. We select the L_1 distance first due to its computational simplicity and second due to its appropriateness for high dimensional data

comparison [166]:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_m |\mathbf{x}_i(m) - \mathbf{x}_j(m)| \tag{6.6}$$

The direct comparison of raw data serves first as a baseline system. Second, it is instrumental in tuning parameters such as the box size, sampling resolution, the type of distance transform function as well as the aperture of the GDT or LDT functions. These optimized parameter settings are then used by all the subspace transform methods. Thirdly, direct comparison method guides us to form a well aligned training set, where coherent axis labels and reflections are selected within classes. This procedure is described in Section 6.5.4.

Calculation of $d(\mathbf{x}_i, \mathbf{x}_j)$ for every pair of the query and target object becomes very time consuming with increasing number of database objects and for large R. It would be inefficient to use the direct comparison method in an online application such as web-based 3D model retrieval. In general, it is desired to have as compact and informative descriptors as possible, without any significant performance loss. In Section 6.5 we investigate subspace methods for compacting features.

6.5. Subspace Methods

Given the observation matrix, $\mathbf{X}$ subspace methods find a set of vectors that describe the significant statistical variations among the observations. These vectors form the basis of a subspace where most of the meaningful information for certain class of processes is preserved, and the orthogonal complement of this space is then considered as noise. The significant part of an observation, $\mathbf{x}$ is expressed as the linear combination of basis vectors $\mathbf{\Phi}$:

$$\mathbf{x} \approx \mathbf{\Phi b} \tag{6.7}$$

where $\mathbf{b} = (\mathbf{\Phi}^H\mathbf{\Phi})^{-1}\mathbf{\Phi}^H\mathbf{x}$. These methods have the additional advantage of greatly reducing the dimensionality of the observations.

Let us assume that we have a set of N training models that are represented as a voxel grid of size $R \times R \times R$. Let $\mathbf{x}_i$ be a column vector of length $M = R^3$ formed by some lexicographic ordering of the voxel values of the i^{th} model. The data matrix is then formed as $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_N]$ and is of size $M \times N$.

6.5.1. Principal Component Analysis

We have described PCA analysis in Section 2.5. However, for the voxel data here, we used noncentered PCA contrary to the common practice. Instead of the covariance matrix, we have calculated the correlation matrix $\mathbf{C} = \mathbf{X}\mathbf{X}^T$ and used the eigenvectors of the correlation matrix. The rest of the procedure is the same as we have explained in Section 2.5 and we re-explain it here: Let $\{\mathbf{u}_1,\ \mathbf{u}_2,\ ...,\ \mathbf{u}_K\}$ be the first K eigenvectors of $\mathbf{C}$ with corresponding eigenvalues $\{\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_K\}$. These vectors model the largest variations among the training samples, therefore are considered to capture most of the significant information. The amount of information maintained depends on K and the spread of eigenvalues. The projection of an input vector $\mathbf{x}$ onto the PCA subspace is given by $\mathbf{a} = \mathbf{U}^T\mathbf{x}$, where $\mathbf{U}$ represents the $M \times K$ projection matrix formed as $[\mathbf{u}_1\ \mathbf{u}_2\ ...\ \mathbf{u}_K]$.

The reason for using the noncentered PCA is that it is much more suitable for data that exhibit high heterogeneity among axes [167]. Each voxel in the 3D grid corresponds to an axis of the vectors of size $M = R^3$. Some classes have negligible activity on some subset of voxels, i.e. axes, since IDT is close to zero for most of the voxels, which are not close to the surface. That is why we consider the data as having high heterogeneity among axes. Choosing non-centered PCA is also validated by our experiments conducted with centered and non-centered data.

Figure 6.6 and Figure 6.7 give visualizations of eigenvectors, or basis shapes, obtained via PCA of the training set of the Princeton Shape Benchmark. The first

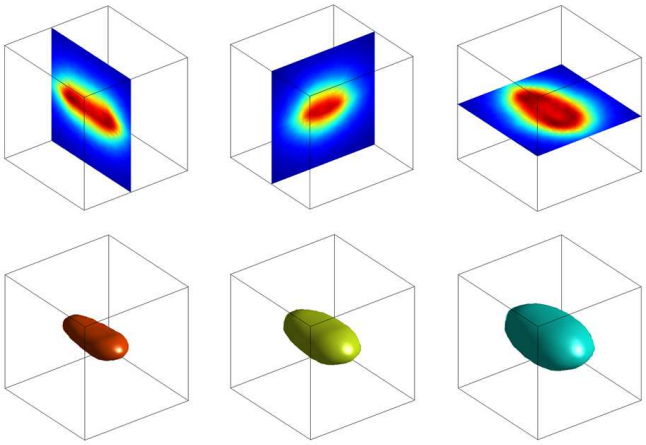


Figure 6.6. Visualization of the first eigenvector. First row show slices from x, y and z axes, from left to right. Second row shows isosurfaces at different levels.

mode of variation is similar to the notion of a "mean shape" (see Figure 6.6). The first coefficient then determines the extent that this "mean shape" is contained in a given shape. Model groups that have a significant non-zero activity only on a subset of voxels can have different means, on top of which the other modes of variations are added. For example elongated and thin models are inactive on most of the outward voxels and they have low projections on the first eigenvector. So they have a thinner ellipsoid (see Figure 6.6) as the first component. The compact and fat objects on the other hand, have a larger ellipsoid as the first component, and other modes of variations carve out the inner parts. Large deviations from the grand-mean of the shapes support our choice for the non-centered PCA.

Figure 6.7 shows the next largest four modes of variations added to the first mode. First mode is multiplied by the first eigenvalue: $\mu = \lambda_1 \mathbf{u}_1$. The reconstructed modes seen in Figure 6.7 are obtained by fixing μ and adding on top of it, the eigenvector of interest weighted by a factor c of the corresponding eigenvalue in the positive and negative directions: $c\lambda_i \mathbf{u}_i + \mu$. Then each of the reconstructed modes is visualized by its slices from three orthogonal directions and also as an isosurface. Note that we have used the IDT while constructing the basis volumes, therefore the eigen-volumes are multi-valued functions in the 3D space.

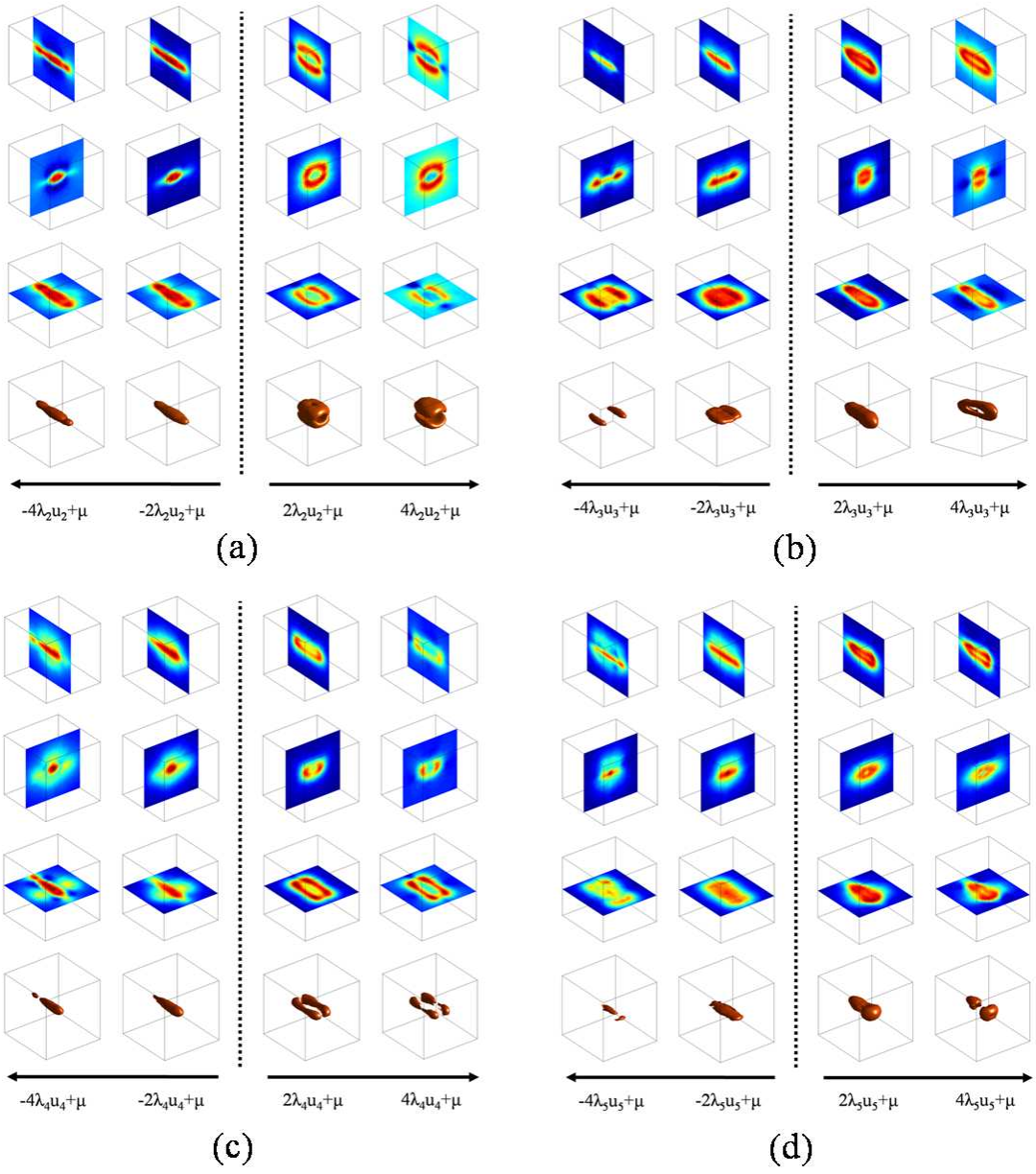


Figure 6.7. Second (a), third (b), fourth (c), and fifth (d) modes of variation. First three rows show slices from x, y and z axes, respectively. Fourth row shows isosurfaces at the same level.

The shape variations in the figures show the effect of each eigenvector. We first observe that the basis shapes are nearly symmetric around some axes. This is due to the symmetric structure of most of the models in the Princeton Shape Benchmark. The projection of any model on these basis shapes will be an indicator of the amount of the corresponding type of symmetry. Our second observation is that the strongest variations are mostly in terms of topological changes. The weights of the eigenvectors account for the formation or disappearance of gross holes and disjoint parts in different shapes.

The second eigenvector controls the elongation of a model (Figure 6.7a). When this coefficient is negative, the model becomes more elongated and thinner, and when it is positive we get a spherical shape with a hole inside. The presence of a hole, rather than a solid sphere, is due to the fact that we have surfaces instead of filled volumes in the training set. Further variation in the positive direction splits the sphere into two parts. The third mode causes the formation of two elongated parts with a negative coefficient, and as the coefficient increases the parts start to merge (Figure 6.7b). An increasing positive third mode coefficient results in a torus. The fourth and fifth modes of variations have similar kind of topological effects (Figure 6.7c and Figure 6.7d). The effects of higher modes become less discernible on the topology and the global shape, and they rather model finer shape variations.

6.5.2. Independent Component Analysis

Since ICA1 behaves like NMF and gives sparse bases similar to those of NMF, it will be more informative to investigate ICA2 architecture as it yields structurally different basis vectors as compared to NMF.

We select K, the reduced dimension obtained by PCA prior to the application of ICA, in a goal-oriented manner, experimentally by observing the retrieval performance over the training set.

Figure 6.8 gives visualizations of ten of the ICA components obtained from

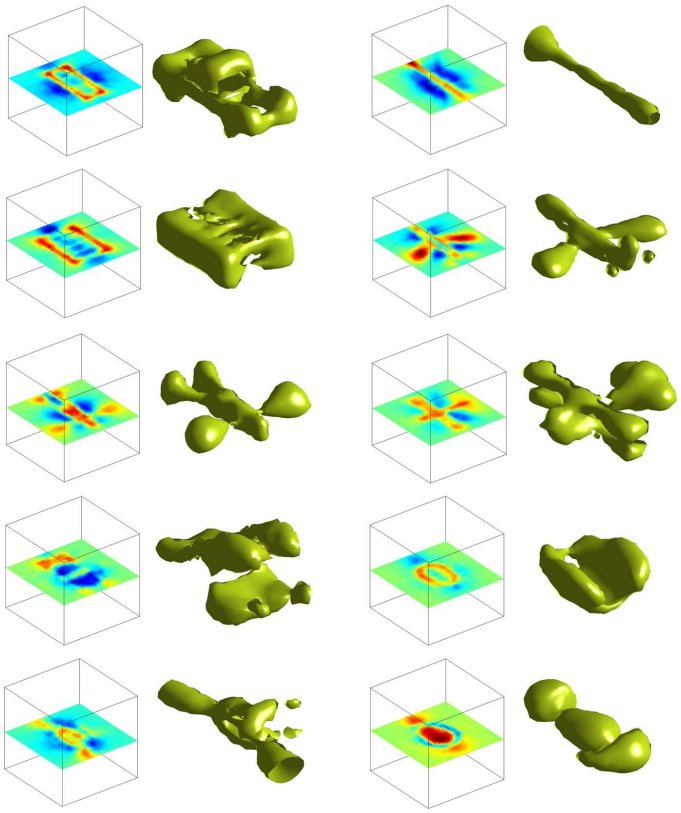


Figure 6.8. Visualization of sample ICA2 basis vectors.

the PSB training set. We visualize each component by a horizontal slice and an isosurface. We have totally different basis volumes from those obtained with PCA. The ICA2 components, or basis volumes, resemble the models that are present in the training set; whereas in PCA we observe very general topological or geometric variations. The PCA projections give clues about the class but distributed over several coefficients. However in the ICA case, whenever a coefficient is pronounced, we have a high correlation or resonance situation giving strong indication of the model class.

6.5.3. Nonnegative Matrix Factorization

We used the multiplicative update rules to get the NMF basis as explained in Section 2.7. As we have mentioned in Section 2.7, only positive bases and coefficients are allowed in NMF. This constraint forces the NMF basis vectors to represent local

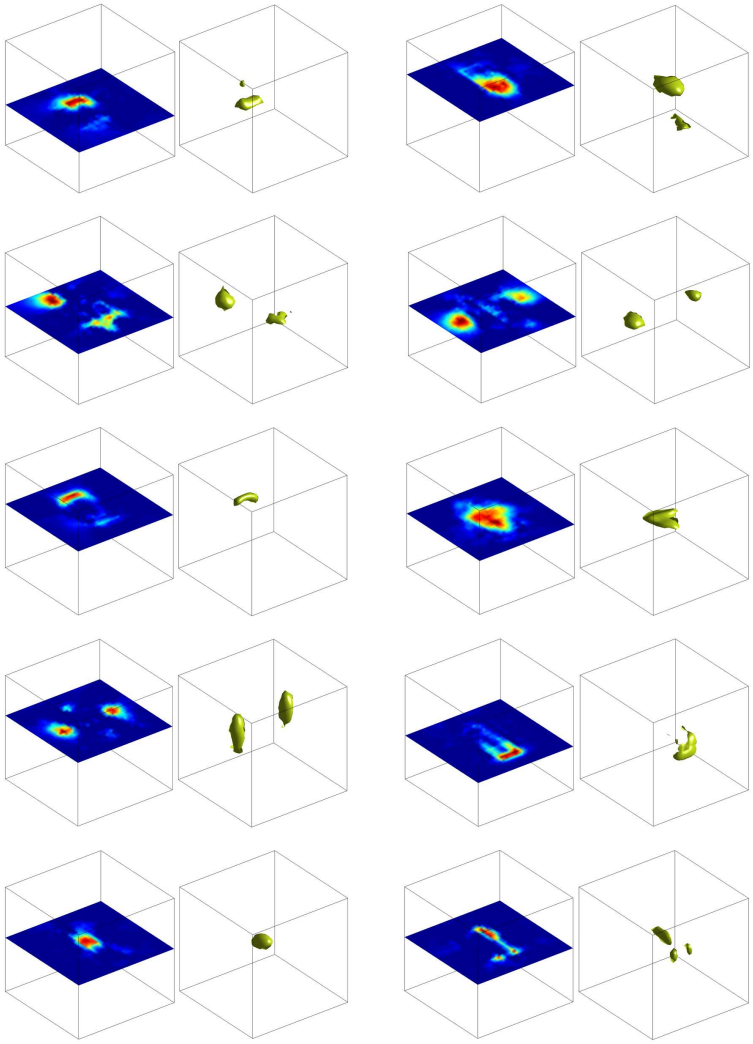


Figure 6.9. Visualization of sample NMF basis vectors.

parts of the models. The visualization of sample NMF components in Figure 6.9 verifies this argument. We get sparse basis volumes representing different parts of the models in the training set.

6.5.4. Axis Relabeling and Reflection

The most problematic issue with the CPCA normalization is the ambiguity of axis ordering and reflections. We conjecture that most of the misalignment errors are due to inconsistent within-class axis orderings and orientations given by the normalization procedure. We will demonstrate this fact in Section 6.7.4 by showing

the non-negligible gains in the retrieval performance when more coherent within-class orientations are available.

We resolve the axis ordering and reflection ambiguities by generating the set of all 48 possible reflections and orientations of the objects. Fortunately these 48 poses are generated very rapidly by applying array transpositions to the voxel-based representation of 3D models. We simply permute the dimensions of the 3D array to alter axes relabeling and flip the array along the three dimensions to obtain reflected representations of the voxel-based models. We will refer collectively to these pose varieties of the voxel-array as 48-Axes Relabeled and Reflected (48-ARR) versions of the model. For the i^{th} model in the database, the r^{th} ARR version will be denoted as $\mathbf{x}_i^r$, with $r = 1,\ 2,\ ...,\ 48$.

While constructing the data matrix at the training stage, we correct the inappropriate axes ordering and orientations by applying one of the following corrective schemes: We find the most appropriate axis labeling and reflection by: (i) Mean shape based ARR selection (MbARR), or (ii) Class based ARR selection (CbARR).

6.5.4.1. Mean Shape Based ARR Selection. This procedure assumes that the training set is not annotated with class information. In this case, we calculate the mean shape $\mathbf{m}$, by averaging the training samples $\{\mathbf{x}_1,\ \mathbf{x}_2,\ ...,\ \mathbf{x}_N\}$. Using direct voxel comparison method, we find the best among the 48-ARR versions of each model as $r_i = \arg\min_{r=1,\,2,\,...,\,48} |\mathbf{x}_i^r - \mathbf{m}|$. Then we recalculate the mean and repeat the procedure iteratively until the mean shape is not altered anymore.

6.5.4.2. Class Based ARR Selection. In the second procedure, we assume that the training set is coupled with the class information of the models. For each class C, we select an arbitrary member $\bar{\mathbf{x}}_C$ as the representative of the class. Then we find the best 48-ARR version of the remaining members of the class via direct comparison of voxels: $r_i = \arg\min_{r=1,\,2,\,...,\,48} |\mathbf{x}_i^r - \bar{\mathbf{x}}_C|$ for $\mathbf{x}_i \in C$.

The assumption of the presence of class information of the samples in the training set is always valid in supervised classification applications of 3D shapes, such as face recognition or detection of pathologies of organs. In database retrieval, on the other hand, annotation of large databases is not always possible. However, a well-sampled subset of the database can be annotated and used to build the subspaces. Most of the systems including web-based search have subsets of annotated models that can be reserved for training.

6.6. Matching

After the subspace is trained and the bases are formed, the target and query models are projected on the subspace, and these projections are used as the shape descriptor. We apply CPCA to the query and target models, voxelize them and define the IDT functions in the 3D space. For each model we also get the 48-ARR versions and project each one onto the subspace. We have a set of feature vectors, $\mathbf{F}_i = \{\mathbf{f}_i^1,\ \mathbf{f}_i^2,\ ...,\ \mathbf{f}_i^{48}\}$ for the i^{th} model defined as:

$$\mathbf{f}_i^r = \mathbf{P}_{\mathbf{\Phi}}\mathbf{x}_i^r \quad r = 1,\ 2,\ ...,\ 48 \tag{6.8}$$

where $\mathbf{P}_{\mathbf{\Phi}}$ is the projection operator for the subspace spanned by $\mathbf{\Phi}$. In order to assess the dissimilarity between two models i and j, we construct the distance matrix, $\mathbf{D}$ of the 48×48 pairings from the two sets, $\mathbf{F}_i$ and $\mathbf{F}_j$, such that

$$\mathbf{D}_{rq}(\mathbf{F}_i, \mathbf{F}_j) = d_c(\mathbf{f}_i^r, \mathbf{f}_j^q) \tag{6.9}$$

where, we use cosine distance to compare pairs of feature vectors:

$$d_c(\mathbf{f}_i^r, \mathbf{f}_j^q) = \frac{\mathbf{f}_i^r \mathbf{f}_j^q}{|\mathbf{f}_i^r||\mathbf{f}_j^q|} \tag{6.10}$$

We have also conducted experiments with L_1 and L_2 distances to compare pairs of feature vectors. However, cosine distance gives the best performance since it

Table 6.1. Pseudo-code for the pose assignment strategy.

Step 1:	Initialize $COST = 0$;
Step 2:	$(\hat{r},\ \hat{q}) = \arg\min_{r,\,q} \mathbf{D}_{rq}$
	$COST \leftarrow COST + \min_{r,\,q} \mathbf{D}_{rq}$
	$\mathbf{D}_{\hat{r}q} \leftarrow \infty \quad q = 1, 2, ..., 48$
	$\mathbf{D}_{r\hat{q}} \leftarrow \infty \quad r = 1, 2, ..., 48$
Step 3:	Stop if all poses (r, q) are assigned to each other (or all $\mathbf{D}_{rq} = \infty$). Otherwise go to Step 2.

normalizes the norms of the feature vectors to unity. After constructing the distance matrix among the feature vectors corresponding to 48-ARR versions of the two models, we either select the minimum of the matrix as the distance between the two models (the MIN rule) or use the following fast variant of the Munkres algorithm, which we call as the pose assignment strategy. We define a cost function of the one-to-one assignment of each 48-ARR version of a model to a 48-ARR version of another model. We initialize the cost to zero. We select the minimum element of the matrix and add its value to the cost function. We set all elements in the row and column of the minimum element to infinity, and search for the next minimum of the distance matrix. We repeat the procedure until all 48-ARR versions of the two models are assigned to each other in a one-to-one manner. The final cost is the distance between the two models. The pose assignment strategy is more robust and it improves the performance significantly as opposed to the MIN rule. In MIN rule we consider only one pair of pose match, while in the pose assignment strategy we use all the distances between matched pose pairs. The pseudo-code for the pose assignment strategy is given in Table 6.1.

6.7. Experimental Results

We have conducted our experiments on the database of Princeton Shape Benchmark [168]. The database consists of a training set with 907 models in 90 classes and a test set with 907 models in 92 classes. The training and test sets are disjoint in

the sense that they do not have common models. Although most shape classes are common to both, each set includes classes not present in the other. We use precision-recall curves, discounted cumulative gain (DCG), nearest neighbor (NN), first tier (FT) and second tier (ST) as measures of retrieval performance. Let C be the class of a query model and $|C|$ be the number of models of class C among the database of target models. Let K be the number of retrieved models and, K_C be the number of models that belong to class C among the K retrieved models. The evaluation measures are defined as follows:

- **Recall:** Given K, recall is the proportion of K_C to $|C|$.
- **Precision:** Given K , precision is the proportion of K_C to K.
- **First tier (FT):** First tier is equal to the recall at $K = |C|$.
- **Second tier (ST):** Second tier is equal to the recall at $K = 2|C|$.
- **Nearest neighbor (NN):** Nearest neighbor is the rank-1 classification accuracy.
- **Discounted cumulative gain (DCG):** To calculate discounted cumulative gain, we obtain a list, G_k of the retrieved models, where G_k is 1 if the k^{th} model belongs to C and, 0 otherwise. Then the DCG at k is equal to

$$DCG_k = \begin{cases} G_k\,, & \text{for } k = 1 \\ DCG_{k-1} + \frac{G_k}{\log_2 k} & \text{for } k = 2,\ 3,\ ...,\ k_{max} \end{cases} \tag{6.11}$$

The overall DCG is calculated as

$$DCG = \frac{DCG_{k_{max}}}{1 + \sum_{k=1}^{|C|} \log_2 k} \tag{6.12}$$

6.7.1. Selection of the Box Size

Prior to voxelization of the models in a database (e.g., PSB), we should set the size of the box in which the models will be rastered. We select the box size by inspecting the statistics of the pose-normalized triangular mesh models in the PSB training set. As explained in Section 6.3.2, we examine the extremities along x, y, and z directions and the surface areas outside the box.

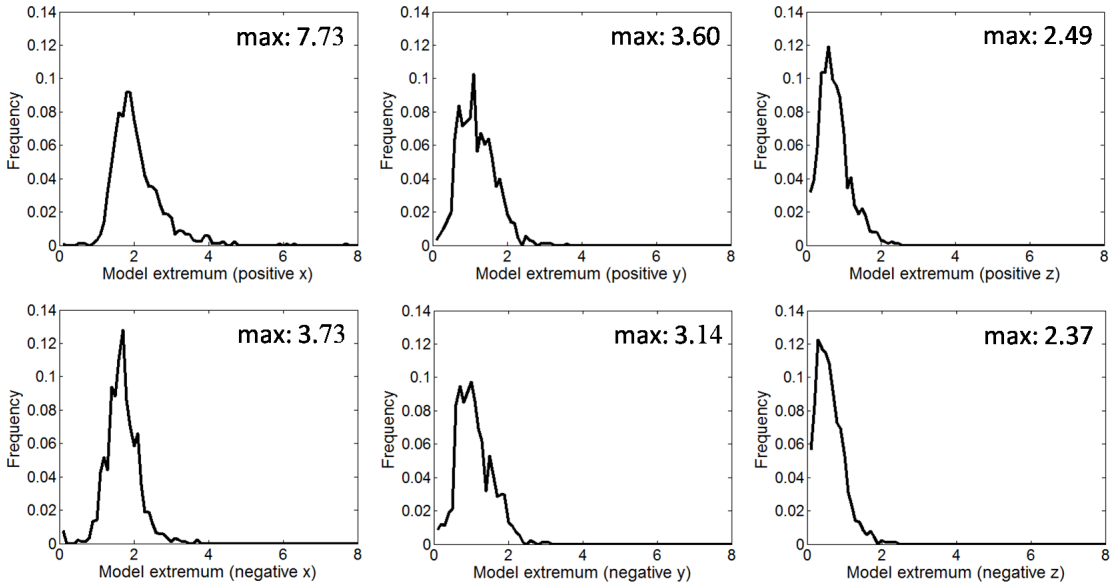


Figure 6.10. Histograms of model extrema along positive and negative x, y, and z directions.

Figure 6.10 shows the histograms of the extremities along positive and negative x, y, and z directions in the PSB training set. The extremities are larger in the x direction since during the PCA normalization an object is aligned such that the orientation along the highest dispersion coincides with the x-axis. The "max" figure in each graph is the maximum extremity encountered among the models in the PSB training set. Inspecting the histograms for y and z, we can safely set the box size to 2.5. However, extremities along the x direction go well beyond 2.5 for many models. Table 6.2 gives the percentage of objects that will be cropped with respect to the choice of the box size. When we select 2.0 for box size, more than half of the objects will be cropped. When the box size is set to 2.5, nearly 25 per cent of the objects will not fit in the box.

We can determine the ratio, a_i, of the cropped surface area of the i^{th} object to the object's total area as a function of box size. Table 6.2 gives the statistics of a_i with respect to the box-size over the PSB training set. We can observe that less than one per cent of the surface of an object will be cropped on average if we select a

Table 6.2. Statistics of cropped models and cropped surface area percentage (a_i) with respect to box size (PSB training set).

Box size	**# cropped models**	**% cropped models**	**Max(a_i)**	**Mean(a_i)**	**Mean(a_i), over cropped models**	**Median(a_i), over cropped models**
2.0	474	52.26	16.62	2.34	4.47	3.82
2.5	211	23.26	8.97	0.50	2.15	1.14
3.0	80	8.82	7.29	0.15	1.69	1.04
3.5	31	3.41	6.44	0.05	1.57	0.88
4.0	13	1.43	4.95	0.02	1.45	0.72
4.5	6	0.66	3.80	0.01	1.92	1.90
5.0	4	0.44	2.63	0.01	1.91	2.22

box size of 2.5. When the average is taken over only the cropped models, the ratio of outside surface area per object is only 2.15 per cent, and the median is half that amount. Based on these observations, we have decided to fix the box size at 2.5 in all the experiments.

6.7.2. Comparison of 3D Distance Functions

In this section, we compare the 3D functions defined in Sections 6.3.2 and 6.3.3 with respect to their retrieval performances on the PSB set. The aim of the experiments in this section is to determine the optimal of these 3D functions without the use of any subspace technique. Instead, we use direct comparisons method described in Section 6.4. The experiments are conducted on the PSB training set. In these experiments we did not calculate 48×48 comparison scores between model pairs; instead we have used only one pose of each model while matching. The pose is either the one given by the CPCA or is determined using CbARR. Table 6.3 gives the NN and DCG values for three resolutions of voxelization; i.e. for $R = 16$, $R = 32$, and $R = 64$. The values under the column entitled as NoARR correspond to the cases without any pose optimization, hence with the use of the pose obtained by CPCA. The values under the CbARR refer to the cases with class-based ARR selection described in 6.5.4.2. We omitted the results for a pose selection using MbARR here,

Table 6.3. Performance of 3D functions for various resolutions on the training set of Princeton Shape Benchmark. Direct comparison method is used.

		R=16		R=32		R=64	
		NoARR	**CbARR**	**NoARR**	**CbARR**	**NoARR**	**CbARR**
NN	**Binary**	47.0	56.4	49.8	58.1	33.8	38.6
	DT	48.1	60.4	51.8	63.9	55.1	67.8
	IDT	**51.0**	**62.3**	**58.0**	**69.1**	**58.3**	**71.1**
	GDT (σ = 1)	49.2	60.2	53.7	63.6	48.3	67.6
	GDT (σ = 3)	49.4	61.6	56.0	66.2	55.5	67.7
	GDT (σ = 6)	47.9	60.4	54.7	64.1	56.7	67.6
	GDT (σ = 10)	47.3	58.7	52.6	64.9	57.2	67.7
	LDT (k = 2)	49.6	59.8	54.7	65.9	50.5	67.9
	LDT (k = 5)	49.7	62.2	56.4	65.5	55.6	67.9
	LDT (k = 10)	48.1	60.1	54.6	64.7	57.3	67.9
DCG	**Binary**	50.5	57.7	50.2	56.9	40.3	44.2
	DT	52.0	61.6	54.8	64.5	55.5	65.6
	IDT	**53.4**	**62.3**	**56.6**	**66.2**	**57.0**	**66.8**
	GDT (σ = 1)	52.0	60.1	53.5	61.7	48.9	64.6
	GDT (σ = 3)	52.5	61.5	55.3	64.3	55.1	64.4
	GDT (σ = 6)	51.5	60.9	55.2	64.1	55.8	64.4
	GDT (σ = 10)	51.1	60.4	54.7	64.4	55.9	64.4
	LDT (k = 2)	52.3	60.5	54.0	62.6	50.6	64.7
	LDT (k = 5)	52.6	61.8	55.3	64.4	55.0	64.7
	LDT (k = 10)	52.0	61.3	55.4	64.2	55.8	64.7

since it gives similar ordering of performance figures among the functions of distance transform.

The most significant result of the experiments is that IDT performs much better than all the other functions at all the three resolutions. The binary function performs poorly as expected. Its performance even deteriorates for increasing voxel resolution, since the finer resolutions result in greater mismatch among similar models. Since the shapes of GDT and LDT profiles are similar, their performances for corresponding apertures are close to each other. At resolution 64, small apertures yield poor performance when we do not use class-based ARR selection. Another observation

is that we get a much bigger increase in NN as compared to DCG, when we increase resolution. This is because fine resolutions favor target models that are very similar to the query model. However, for target objects in the same class but do not have well matching details with the query model, an increase in resolution may not raise their ranks. With IDT, we gain two points for NN and 0.6 points for DCG when the voxel resolution goes from 32 to 64. These observations have lead us to adopt $R = 32$ for resolution in all the following experiments.

6.7.3. Performance Analysis of Subspace Methods

6.7.3.1. Training Phase. In order to select the dimensionality of the PCA, ICA and NMF subspaces, we perform experiments on the training set of Princeton Shape Benchmark. We either leave the training set without any pose correction (NoARR) or apply mean shape-based (MbARR) or class-based ARR selection (CbARR), the latter two with the goal of selecting the best representative of 48-ARR versions of each model in the training set. Once the subspaces and their basis vectors are obtained, we extract the feature vectors corresponding to the 48-ARR versions of each model in the training set. We apply the MIN rule (Section 6.6) to match the sets of feature vectors of the query and target models for the NoPC and MbPC cases. However, for the CbPC case, we directly use the best representative of 48-ARR versions of each model and do not perform 48×48 comparisons between query and target models.

Figure 6.11a, b and c show the DCG versus dimension curves obtained with PCA, ICA and NMF, respectively. For all subspaces, class-based ARR selection boosts the retrieval performance; since we greatly reduce the 90 degrees pose ambiguities within classes. The performance of PCA remains robust with respect to increasing dimension, since higher order PCA coefficients have lower impact on the similarity of the models. With ICA, the performance is quite sensitive to the dimensionality. We have a peak performance at dimension 40, regardless of the alignment scheme of the training models (Figure 6.11b). NMF-based retrieval scheme yields stable results with increasing dimension as compared to ICA, although the DCG values fluctuate a little due to random initialization of NMF basis vectors. With these observations,

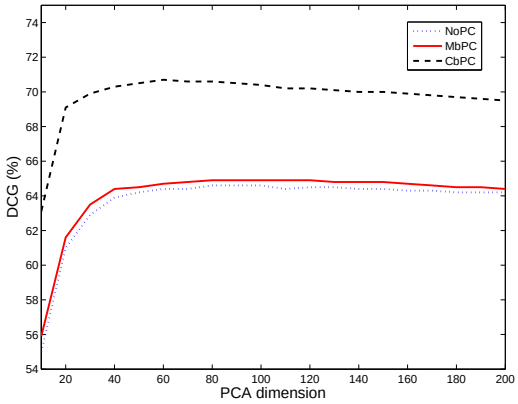
Table 6.4. Retrieval performances on the PSB test set. The pose correction is only performed on the PSB training set during the subspace building phase. MIN rule is used to match query and target models.

Subspace	Dimension	Pose correction	NN	FT	ST	DCG
PCA	100	MbARR	61.2	58.0	57.1	61.6
		CbARR	61.7	58.4	56.8	61.5
ICA	40	MbARR	57.8	56.2	53.6	59.8
		CbARR	58.4	55.2	52.9	59.2
	100	MbARR	62.1	59.2	56.8	61.2
		CbARR	62.6	59.8	57.0	61.4
NMF	70	MbARR	61.0	58.1	55.7	61.1
		CbARR	60.3	58.9	56.3	60.7
	100	MbARR	62.0	60.0	56.4	61.5
		CbARR	61.7	59.1	56.9	61.0

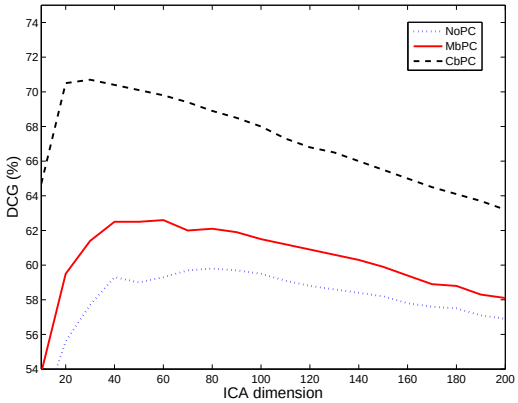
we set the dimension of PCA-based subspace to 100 for the PSB test set experiments. For ICA, we set the dimension either to 40, following the peak of DCG with the training set or to 100 in order to have the same dimension with PCA. Likewise, for the NMF-based experiments, we report results with dimensions 70 and 100.

6.7.3.2. Performances on PSB Test Set. Regardless of which ARR selection scheme we have used in the training phase, we do not use any class information in the experiments conducted over the test set. In Table 6.4 we give the retrieval performances of the three subspaces obtained on PSB test set, with the MIN rule, whereas the results in Table 6.5 are obtained using pose assignment strategy (Section 6.6). Clearly, pose assignment strategy provides a significant gain to the performance.

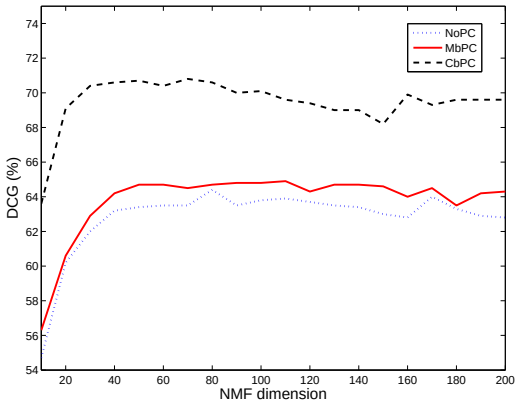
Figure 6.12 gives the precision-recall curves for the three subspace methods. The curves correspond to the case where we use the pose assignment strategy. When we compare the three subspaces, we can observe that the performance of PCA-based scheme is lower than the ICA and NMF-based schemes. ICA and NMF subspaces give comparable results, although ICA performs slightly better. We can



(a)



(b)



(c)

Figure 6.11. DCG versus subspace dimension with PCA (a), ICA (b) and NMF (c). Experiments are conducted on the PSB training set.

Table 6.5. Retrieval performances on the PSB test set. The pose correction is only performed on the PSB training set during the subspace building phase. Pose assignment strategy is used to match query and target models.

Subspace	Dimension	Pose correction	NN	FT	ST	DCG
PCA	100	MbARR	63.2	37.1	48.1	63.4
		CbARR	63.5	37.0	48.2	63.4
ICA	40	MbARR	66.2	38.4	51.2	65.0
		CbARR	66.4	38.5	50.7	64.8
	100	MbARR	66.5	39.5	51.4	65.5
		CbARR	66.5	39.4	51.5	65.6
NMF	70	MbARR	66.3	38.6	50.3	64.9
		CbARR	66.9	38.5	50.4	64.7
	100	MbARR	66.8	39.0	50.7	65.0
		CbARR	66.9	38.7	50.0	65.0

see that the class-based ARR selection of the training set brings almost no gain to the performance on the test set. Another disparity between the training and test cases is about the dimension. Although the performance drops after the ICA dimension of 40 with the retrieval experiments on the training set (Figure 6.11b), when we switch to the test set, we have performance gains with a higher dimension. Some classes in the PSB test set are not present in the training set, therefore a well-tuning of the parameters with the training set does not necessarily reflect on the test set. However, an inspection of Table 6.5 reveals that we do not have dramatic dependency on the selection of the pose correction scheme or the dimension. So we do not need to have a labeled training set to incorporate class-based alignment while constructing the subspace models.

6.7.4. The Correct Pose

For the sake of emphasizing the importance of the within-class coherence of axes labeling and reflection, we give performance results on the PSB test set, assuming that the best 48-ARR version of each model is known. We perform a class-based pose-correction on the models of the test set using direct comparisons method. We

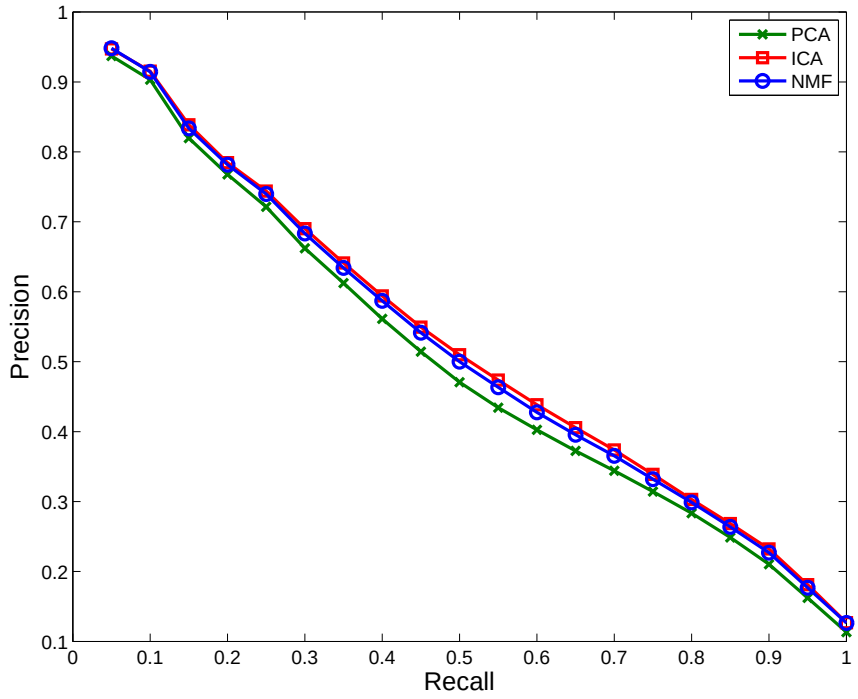


Figure 6.12. Precision-recall curves on the PSB test set. Mean shape based pose correction is applied to the training set. Pose assignment strategy is used to match query and target models.

obtain the 48-ARR versions of the IDT representation of a model and select the one that gives the least L_1 error with the class representative (Figure 6.13). This is a hypothetical case where we assume that the axes of each model are correctly labeled. Table 6.6 gives the performance of various descriptors with this ideal case. We can observe the boost in the performance when we compare the results with those in Table 6.5, and this comparison shows that coherent axes labeling is crucial when PCA normalization is applied to the models. With the pose assignment strategy, we try to achieve the ideal results presented in Table 6.6.

6.7.5. Comparison with State of the Art

In order to demonstrate the potential of subspace techniques for 3D model retrieval, we compare our results to the state-of-the-art methods. We select the

Table 6.6. Retrieval performances on the PSB test set assuming that correct axis labeling and reflection of each model are known.

Subspace	Dimension	NN	FT	ST	DCG
PCA	100	70.0	43.0	53.8	68.3
ICA	40	69.6	43.7	55.2	68.6
	100	72.4	43.4	53.9	68.9
NMF	70	70.3	43.7	54.9	68.9
	100	71.4	43.8	55.1	69.2

Figure 6.13. Class-based ARR selection for bench seat and rectangular table classes. The top red figures are the reference models. Pink figures at the left are the outputs of CPCA-based normalization. Cyan figures under the reference models are the best choice out of the 48-ARR representations.

Table 6.7. Comparison of subspace methods with the state of the art 3D shape descriptors on PSB test set.

Descriptor	NN	FT	ST	DCG
CRSP	67.9	40.5	52.8	66.8
DSR	66.5	40.3	51.2	66.5
DBF	68.6	39.3	50.0	65.9
ICA	66.5	39.5	51.4	65.5
NMF	66.8	39.0	50.7	65.0
LFD	65.7	38.0	48.7	64.3
PCA	63.2	37.1	48.1	63.4

four top performing methods that were evaluated in [152, 153], namely, concrete radialized spherical projection (CRSP) [154], DSR descriptor [128], density based framework (DBF) [152, 153], and light field descriptor (LFD) [169]. The CRSP scheme decomposes the models into a set of spherical functions, which are then encoded using spherical harmonics. The DSR descriptor is a hybrid descriptor that combines depth buffer and silhouette-based descriptors. The DBF characterizes models using multivariate probability density functions of local surface features. The LFD is a collection of views of an object from uniformly sampled points on a sphere. With the exception of LFD, the three methods use CPCA for pose-normalization. In CRSP, the normalization is even enhanced with another PCA normalization that is based on surface normals. None of the four descriptors employ learning schemes that use class information of training or target models. Similarly we abstain from any class information and use unsupervised ARR selection based on the mean shape of the training database (Section 6.5.4).

Table 6.8 gives performance results obtained with fusion of the subspace methods with each other and with the DSR and DBF descriptors. The fusion is performed via the summation of the scores obtained from each descriptor. The fusion of subspace methods with each other does not bring much gain, except the NN measure with the fusion of ICA and NMF. However, when we combine the ICA and NMF-based descriptors with the DSR or DBF, we get a significant improvement of retrieval

Table 6.8. Fusion of subspace methods with other descriptors.

Fusion	**NN**	**FT**	**ST**	**DCG**
PCA+ICA	67.0	39.7	51.3	65.6
PCA+NMF	65.9	38.8	50.6	64.9
ICA+NMF	67.3	39.4	51.3	65.5
PCA+ICA+NMF	66.7	39.6	51.2	65.5
Fusion with DSR	**NN**	**FT**	**ST**	**DCG**
DSR+PCA	67.7	41.6	53.1	67.4
DSR+ICA	69.3	44.2	55.4	69.1
DSR+NMF	69.1	43.8	55.1	68.8
DSR+ICA+NMF	71.0	44.7	56.1	69.6
Fusion with DBF	**NN**	**FT**	**ST**	**DCG**
DBF+PCA	69.2	40.5	51.4	66.8
DBF+ICA	70.5	42.5	53.7	68.2
DBF+NMF	70.2	41.9	53.0	67.8
DBF+NMF+ICA	70.1	43.3	54.6	68.7
Fusion with DBF and DSR	**NN**	**FT**	**ST**	**DCG**
DBF+DSR	73.4	45.0	56.2	70.2
DBF+DSR+NMF+ICA	73.6	46.2	57.7	71.1

performance. These results show that, the subspace methods can be even more beneficial when used in combination with other methods. Indeed, when we combine the DSR, DBF, ICA and NMF methods, we achieve the highest performance reported so far on the PSB test set, among the unsupervised retrieval methods in the literature.

6.8. Conclusion

In this chapter, we have developed 3D model retrieval schemes using various subspaces of object shapes. We have investigated the potential of three popular techniques, PCA, ICA and NMF, since each of them describes somewhat different statistical characteristics of the data. Being reconstructive methods, these features can easily gloss over minor differences and defects, but are affected by the gross pose uncertainties. The pose dependency of the subspace methods is solved by the use of CPCA-based pose normalization, followed by voxelization, inverse distance

transform and exhaustive pose optimization. The two main results of our research is, first that ICA and NMF-based schemes provide retrieval performances on a par with the alternate state-of-the-art methods, and second that decision fusion of these schemes advance the performance on Princeton Shape Benchmark beyond that of any one method.

We conjecture that there is still room for performance improvement. Our future research effort will concentrate on the following:

- The subspace methods can be applied on alternative representations of the data, for example on the point cloud or the depth image representations instead of the voxel data.
- Robust versions of subspace building can offer enhanced solutions, especially when the data is corrupted by outliers. In this respect, kernel PCA, sparse PCA [170, 171], robust PCA [172] or other variants of ICA and NMF can be adopted and compared.
- The matching strategy can be improved by considering finer pair-wise alignment of models. One can consider matching manifolds of projections, obtained by fine sampling of the rotation space instead of simply using the possible mirror reflections and axis re-labelings.

7. CONCLUSIONS

In this thesis, we use various subspace-based techniques for three different object recognition applications: (i) Hand biometry, (ii) 3D face biometry, and (iii) Indexing and retrieval of general 3D models. Our main contribution has been to devise various combinations of object representations and subspace methods to optimize the performance. In addition, we introduced normalization, alignment and correspondence building techniques specific to each of the above problems.

7.1. Subspace Analysis

The human visual system operates in a subspace: The high dimensional optical information arriving to the eyes are projected onto a three-dimensional color subspace. The axes of this subspace are determined by the frequency response of the retinal cones. A change of coordinates takes place at an early stage in vision: In fact, the scenes observed by humans are not represented in terms of the actual frequencies of the visible light but a linear combination of the frequency responses of the retinal pigments.

Alternative coordinate systems other than the native one are chosen so that we can optimally model the significant variations of the data, and we can perceive ”patterns” in the data better. Another relevant benefit is the potential reduction in signal dimensionality by capturing the apparent degrees of freedom (or intrinsic dimensionality) of the observed data. For example, face scans represented as high-dimensional point sets belong to a manifold of intrinsically very low dimension. This is also true for hand images and 3D models represented in high-dimensional pixel or voxel arrays.

7.2. Choice of Subspaces

There are a number of criteria to choose a subspace, such as minimum approximation error, least representation entropy, uncorrelatedness, maximum variance of coefficients, statistical independence, sparsity of basis functions, sparsity of coefficients, maximum separation of classes, and so on. In this thesis, we have relied on experimental results to judge for the appropriateness of any specific subspace method for a particular application. However, there are some general clues that can be referred before experimentation, and can be validated through experimentation:

- If sufficient training data are not available, model-driven subspaces such as DFT and DCT are preferrable.
- DFT and DCT-based feature extraction techniques yield parsimonious representations for signals with compact spectral support. Hence if the spatial organization of the data has smoothly varying characteristics, i.e. the spectrum of the data is concentrated at low frequencies, DFT or DCT may be suitable.
- PCA and ICA-2 assume integrity of the shape structure, hence they respond poorly when the input shape is partial. In our discussions throughout the thesis, we have assumed non-occluded, complete shapes. However, if the inputs happen to be partial, application of NMF may be more beneficial [173, 174].
- LDA and QR-decomposition (a half-way to LDA) are highly beneficial if there are enough samples from the classes and the samples well represent the inter-class and intra-class variations. We have especially observed their effectiveness in the case of 3D face recognition.
- If the subspaces built by PCA, ICA and NMF sufficiently reconstruct the data (e.g. the energy is mostly conserved), application of LDA on top of them will redefine the axes of the final subspace according to class separability. It has been observed that the three subspaces, when incorporated with LDA, perform similar to each other.
- For hand recognition, our normalization algorithm turns out to be very successful in suppressing intra-class shape variations. Thus, LDA is only necessary The ICA-based scheme yields over 99 per cent correct identification rate with-

out LDA. However, LDA is beneficial for geometric measures of very different genres (areas, widths, perimeters) or for noisy contour information.

7.3. Contributions of the Thesis

This thesis involves advances and assessment of registration, feature extraction and classification techniques as applied to hand images, 3D faces and 3D generic objects. The main focus was the advancement of the recognition performance in each application beyond the state-of-the-art via subspace techniques. The common strategy for dealing with these various types of signals consists of two stages: (i) Preprocessing of input measurements with emphasis on registration, (ii) Effective application of subspace tools for feature extraction and classification.

The effectiveness of the subspace-based tools relies largely on the success of the pose-normalization and correspondence building steps. This requirement is also true for any other pose-dependent feature extraction method. However, subspace techniques, especially the data-driven ones, are sensitive to intra-class pose and scale variations and incorrect correspondences. Since 2D hand, 3D face and 3D generic objects present different signal characteristics, alignment and pose normalization procedures appropriate for each case were devised.

Our hand normalization algorithm is extremely robust and can compensate for all realistic geometric deformations of the hand. This has opened the way to the use of several subspace-based methods on the contours, silhouettes or texture of the hand. Our identification and verification performances seems to be by far the best as compared to the methods in the open literature. Furthermore, our hand database is an order of magnitude larger than any other publicly available hand database.

We have shown that subspace-based methods applied on various representation modalities of 3D faces provide satisfactory results and that there is still room for improvement via fusion schemes. The subspace features can model the inter-class variations well if there is enough training data with the incorporation of LDA or

QR-decomposition to reweight the subspace coefficients. In order to mitigate expression variations, we have suggested planar warping and masking schemes and we have obtained some improvement. However, expression variation still remains a challenge.

The application of data-driven subspace techniques was not previously considered for the indexing and retrieval of 3D models since correspondence building proved too difficult among different genres of objects. For example, it is nearly impossible to align a 3D cat model onto a biplane model. In this thesis, we have attacked this difficulty at the preprocessing stage by using pose normalization, voxelization and distance transform, and at the matching stage by considering the subspace projections of a number of rotated versions of a model.

7.4. Challenges and Future Work

One can note the discrepancy between performance scores of subspace methods in the three applications, which varied from 99 per cent for hands to 96 per cent for faces and 70 per cent for generic objects. Obviously, the hand space has a low dimensionality and hands are well registered. Faces, on the other hand, have more confounding factors like expressions, they are inherently noisier, and their subspace dimensionality is higher. Finally, generic 3D object database proves the hardest to register, and the variety of object shapes indicate to a much larger dimensionality. In fact, other state-of-the-art methods for 3D model retrieval do not give any higher performance, in other words, subspace methods give results comparable to the state-of-the art methods, which are not using subspace notion.

In biometry, the notion of similarity is well-defined: The biometric measurements belong to a person or not. Even if the hand images of two identical twins may seem extremely similar in shape, a successful recognition system should consider them as different. The content-based generic model retrieval, on the other hand, is an ill-posed problem: There may not be a well-defined ground truth for the categorization of the objects; i.e. the categorization is confined to be subjective and specific

to a particular application. Furthermore, the semantic gap between the linguistic descriptions of the objects and their measurable shape characteristics may become too wide.

The quest for robust, fast and reliable pose normalization or alignment tools for generic objects is still a challenge. The current trend in 3D model retrieval is to use a number of pose normalization algorithms together. We plan to address this challenge in two different ways: (i) Building fast correspondences among 3D models to enable accurate alignment. (ii) Building manifolds of target models by obtaining various rotated versions of the model.

REFERENCES

1. Kendall, D. G., D. Barden, T. K. Carne, and H. Le, *Shape and Shape Theory*, Wiley, 1999.

2. Loncaric, S., "A Survey of Shape Analysis Techniques", *Pattern Recognition*, Vol. 31, pp. 983–1001, 1998.

3. Costa, L. D. F. and R. M. Cesar, *Shape Analysis and Classification: Theory and Practice*, CRC, 2000.

4. Besl, P. J. and R. C. Jain, "Three-Dimensional Object Recognition", *ACM Computing Surveys*, Vol. 17, No. 1, pp. 75–145, 1985.

5. Donoho, D., "High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality", Lecture delivered at the conference "Math Challenges of the 21st Century" held by the American Math. Society, August 2000.

6. Fisher, R., "The Statistical Utilization of Multiple Measurements", *Annals of Eugenics*, pp. 376–386, 1938.

7. Turk, M. and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71–86, 1991.

8. Hallinan, P. W., "A low-dimensional lighting representation of human faces for arbitrary lighting conditions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 995–999, 1994.

9. Murase, H. and S. K. Nayar, "Illumination Planning for Object Recognition Using Parametric Eigenspaces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 12, pp. 1219–1227, 1994.

10. Murase, H. and S. K. Nayar, "Visual Learning and Recognition of 3-D Objects

from Appearance", *International Journal of Compututer Vision*, Vol. 14, No. 1, pp. 5–24, 1995.

11. Ramamoorthi, R. and P. Hanrahan, "On the Relationship between Radiance and Irradiance: Determining the Illumination from Images of a Convex Lambertian Object", *Journal of Optical Society of America A*, Vol. 18, No. 10, pp. 2448–2459, 2001.

12. Ramamoorthi, R., "Analytic PCA Construction for Theoretical Analysis of Lighting Variability in Images of a Lambertian Object", *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 24, No. 10, pp. 1322–1333, 2002.

13. Cootes, T. F., C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models: Their Training and Application", *Computer Vision and Image Understanding*, Vol. 61, No. 1, pp. 38–59, 1995.

14. Cootes, T. F., G. J. Edwards, and C. J. Taylor, "Active Appearance Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, pp. 681–685, 2001.

15. Goodall, C., "Procrustes Methods in the Statistical Analysis of Shape", *Journal of the Royal Statistical Society B*, Vol. 53, No. 2, pp. 285–339, 1989.

16. Bookstein, F. L., "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 6, pp. 567–585, 1989.

17. Jain, A. K., R. P. W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review", *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 4–37, 2000.

18. Draper, B. A., K. Baek, M. S. Bartlett, and J. R. Beveridge, "Recognizing Faces with PCA and ICA", *Computer Vision and Image Understanding*, Vol. 91, No. 1-2,

pp. 115–137, 2003.

19. Ekenel, H. K. and B. Sankur, "Feature Selection in the Independent Component Subspace for Face Recognition", *Pattern Recognition Letters*, Vol. 25, No. 12, pp. 1377–1388, 2004.

20. Hyvärinen, A. and E. Oja, "Independent Component Analysis: Algorithms and Applications.", *Neural Networks*, Vol. 13, No. 4-5, pp. 411–430, 2000.

21. Yörük, E., E. Konukoglu, B. Sankur, and J. Darbon, "Shape-Based Hand Recognition", *IEEE Transactions on Image Processing*, Vol. 15, No. 7, pp. 1803–1815, 2006.

22. Lee, D. D. and H. S. Seung, "Learning the Parts of Objects by Nonnegative Matrix Factorization", *Nature*, Vol. 401, pp. 788–791, 1999.

23. Lee, D. and H. Seung, "Algorithms for Nonnegative Matrix Factorization", *Advances in Neural Information Processing Systems*, Vol. 13, 2001.

24. Miller, R. P., "Finger Dimension Comparison Identification System", U.S. Patent No. 3576538, 1971.

25. Ernst, R. H., "Hand ID System", U.S. Patent No. 3576537, 1971.

26. Jacoby, I. H., A. J. Giordano, and W. H. Fioretti, "Personnel Identification Apparatus", U.S. Patent No. 3648240, 1972.

27. Sidlauskas, D. P., "3D Hand Profile Identification Apparatus", U.S. Patent No. 4736203, 1988.

28. Gunther, M., "Device for Identifying Individual People by Utilizing the Geometry of their Hands", Eur. Patent No. DE10113929, 2002.

29. Miller, B., "Vital Signs of Identity", *IEEE Spectrum*, Vol. 31, No. 2, pp. 22–30,

1994.

30. Zunkel, R. L., "Hand Geometry Based Verification", *Biometrics*, Vol. 31, No. 2, pp. 87–101, 1999.

31. Jain, A. K., A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 1, pp. 4–20, 2004.

32. Holmes, J., L. Wright, and R. Maxwell, "A Performance Evaluation of Biometric Identification Devices", Technical report, Sandia National Laboratories, 1991.

33. Kukula, E. and S. Elliott, "Implementation of Hand Geometry: An Analysis of User Perspectives and System Performance", *IEEE Aerospace and Electronic Systems Magazine*, Vol. 21, No. 3, pp. 3–9, March 2006.

34. Yörük, E., H. Dutağacı, and B. Sankur, "Hand Biometrics", *Image and Vision Computing*, Vol. 24, No. 5, pp. 483–497, 2006.

35. Dutağaci, H., B. Sankur, and E. Yörük, "Comparative Analysis of Global Hand Appearance-Based Person Recognition", *Journal of Electronic Imaging*, Vol. 17, No. 1, p. 011018, 2008.

36. Pavlovic, V. I., R. Sharma, and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 677–695, 1997.

37. Kuch, J. J. and T. S. Huang, "Vision Based Hand Modeling and Tracking for Virtual Teleconferencing and Telecollaboration", *Proceedings of the Fifth International Conference on Computer Vision*, p. 666, IEEE Computer Society, Washington, DC, USA, 1995.

38. Lin, J., Y. Wu, and T. S. Huang, "Modeling the Constraints of Human Hand Motion", *Proceedings of the Workshop on Human Motion*, p. 121, IEEE Computer

Society, 2000.

39. Jain, A., A. Ross, and S. Pankanti, "A prototype hand geometry-based verification system", *Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication*, 1999.

40. Sanchez-Reillo, R., C. Sanchez-Avila, and A. Gonzalez-Marcos, "Biometric Identification through Hand Geometry Measurements", *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 22, No. 10, pp. 1168–1171, 2000.

41. Wong, R. L. N. and P. Shi, "Peg-Free Hand Geometry Recognition Using Hierarchical Geometry and Shape Matching", *IAPR Workshop on Machine Vision Applications*, pp. 281–284, 2002.

42. Bulatov, Y., S. Jambawalikary, P. Kumarz, and S. Sethiay, "Hand Recognition Using Geometric Classifiers", *DIMACS Workshop on Computational Geometry*, pp. 14–15, 2002.

43. Oden, C., A. Ercil, and B. Buke, "Combining Implicit Polynomials and Geometric Features for Hand Recognition", *Pattern Recognition Letters*, Vol. 24, No. 13, pp. 2145–2152, 2003.

44. Kumar, A., D. C. M. Wong, H. C. Shen, and A. K. Jain, "Personal Verification Using Palmprint and Hand Geometry Biometric", *AVBPA*, pp. 668–678, 2003.

45. Kumar, A., D. C. M. Wong, H. C. Shen, and A. K. Jain, "Personal Authentication Using Hand Images", *Pattern Recognition Letters*, Vol. 27, No. 13, pp. 1478–1486, 2006.

46. Kumar, A. and D. Zhang, "Personal Recognition Using Hand Shape and Texture", *IEEE Transactions on Image Processing*, Vol. 15, No. 8, pp. 2454–2461, 2006.

47. Jain, A. K. and N. Duta, "Deformable Matching of Hand Shapes for Verification", *Proceedings of the IEEE International Conference on Image Processing*, pp.

857–861, 1999.

48. http://www.rcgravel.com/palm_print_identification.htm.

49. Shu, W. and D. Zhang, "Palmprint Verification: An Implementation of Biometric Technology", *Proceedings of the 14th International Conference on Pattern Recognition*, Vol. 1, p. 219, IEEE Computer Society, 1998.

50. Zhang, D. and W. Shu, "Two Novel Characteristics in Palmprint Verification: Datum Point Invariance and Line Feature Matching", *Pattern Recognition*, Vol. 32, No. 4, pp. 691–702, 1999.

51. Duta, N., A. K. Jain, and K. V. Mardia, "Matching of Palmprints", *Pattern Recognition Letters*, Vol. 23, No. 4, pp. 477–485, 2002.

52. Liu, L. L., D. Zhang, and K. Wang, "Palm-Line Detection", *Proceedings of the IEEE International Conference on Image Processing*, pp. 269–272, 2005.

53. Wu, X., K. Wang, and D. Zhang, "A Novel Approach of Palm-Line Extraction", *Proceedings of the Third International Conference on Image and Graphics*, pp. 230–233, 2004.

54. You, J., W. Li, and D. Zhang, "Hierarchical Palmprint Identification via Multiple Feature Extraction", *Pattern Recognition*, Vol. 35, No. 4, pp. 847–859, 2002.

55. Laws, K. I., "Texture Energy Measures", *Proceedings of Image Understanding Workshop*, 1979.

56. Chen, J., C. Zhang, and G. Rong, "Palmprint Recognition Using Crease", *IEEE International Conference on Image Processing*, pp. 234–237, 2001.

57. Funada, J., N. Ohta, M. Mizoguchi, T. Temma, K. Nakanishi, A. Murai, T. Sugiuchi, T. Wakabayashi, and Y. Yamada, "Feature Extraction Method for Palmprint Considering Elimination of Creases", *International Conference on Pattern*

Recognition, Vol. 2, p. 1849, 1998.

58. Wu, X., K. Wang, and D. Zhang, "Fuzzy Directional Element Energy Feature (FDEEF) Based Palmprint Identification", *Proceedings of the 16th International Conference on Pattern Recognition*, Vol. 1, p. 10095, 2002.

59. Wu, X., K. Wang, and D. Zhang, "Palmprint Recognition Using Directional Line Energy Feature", *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 4, pp. 475–478, 2004.

60. Han, C.-C., H.-L. Cheng, C.-L. Lin, and K.-C. Fan, "Personal Authentication Using Palmprint Features", *Pattern Recognition*, Vol. 36, No. 2, pp. 371–381, 2003.

61. Li, F., M. Leung, and X. Yu, "Palmprint Identification Using Hausdorff Distance", *IEEE International Workshop on Biomedical Circuits and Systems*, pp. S3/3–S5–8, Dec. 2004.

62. Kong, A. W.-K., D. Zhang, and W. Li, "Palmprint Feature Extraction Using 2-D Gabor Filters", *Pattern Recognition*, Vol. 36, No. 10, pp. 2339–2347, 2003.

63. Zhang, D., W.-K. Kong, J. You, and M. Wong, "Online Palmprint Identification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 9, pp. 1041–1050, 2003.

64. Kong, A. W.-K. and D. Zhang, "Competitive Coding Scheme for Palmprint Verification", *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 1, pp. 520–523, 2004.

65. Kong, A., D. Zhang, and M. Kamel, "Palmprint Identification Using Feature-Level Fusion", *Pattern Recognition*, Vol. 39, No. 3, pp. 478–487, 2006.

66. Li, W., D. Zhang, and Z. Xu, "Palmprint Identification by Fourier Transform", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 16, No. 4,

pp. 417–432, 2002.

67. Kumar, A. and H. C. Shen, "Palmprint Identification Using PalmCodes", *Proceedings of the Third International Conference on Image and Graphics*, pp. 258–261, 2004.

68. Kumar, A. and D. Zhang, "Personal Authentication Using Multiple Palmprint Representation", *Pattern Recognition*, Vol. 38, No. 10, pp. 1695–1704, 2005.

69. Lu, G., D. Zhang, and K. Wang, "Palmprint Recognition Using Eigenpalms Features", *Pattern Recognition Letters*, Vol. 24, No. 9-10, pp. 1463–1467, 2003.

70. Connie, T., A. T. B. Jin, M. G. K. Ong, and D. N. C. Ling, "An Automated Palmprint Recognition System", *Image and Vision Computing*, Vol. 23, No. 5, pp. 501–515, 2005.

71. Jiang, W., J. Tao, and L. Wang, "A Novel Palmprint Recognition Algorithm Based on PCA&FLD", *Proceedings of the International Conference on Digital Telecommunications*, p. 28, 2006.

72. Pang, Y. H., T. Connie, A. Jin, and D. Ling, "Palmprint Authentication with Zernike Moment Invariants", *Proceedings of the Third IEEE International Symposium on Signal Processing and Information Technology*, pp. 199–202, 2003.

73. Dai, Q., N. Bi, D. Huang, D. Zhang, and F. Li, "M-band Wavelets Application to Palmprint Recognition Based on Texture Features", *Proceedings of IEEE International Conference on Image Processing*, pp. 893–896, 2004.

74. Zhang, L. and D. Zhang, "Characterization of Palmprints by Wavelet Signatures via Directional Context Modeling", *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 34, No. 3, pp. 1335–1347, 2004.

75. Han, C.-C., "A Hand-Based Personal Authentication Using a Coarse-to-Fine Strategy", *Image and Vision Computing*, Vol. 22, No. 11, pp. 909–918, 2004.

76. Lu, G. M., K. Q. Wang, and D. Zhang, "Wavelet Based Independent Component Analysis for Palmprint Identification", *Proceedings of International Machine Learning Conference*, Vol. 6, p. 35473550, 2004.

77. Shang, L., D.-S. Huang, J.-X. Du, and C.-H. Zheng, "Palmprint Recognition Using FastICA Algorithm and Radial Basis Probabilistic Neural Network", *Neurocomputing*, Vol. 69, No. 13-15, pp. 1782–1786, 2006.

78. Hennings, P. and B. V. K. V. Kumar, "Palmprint Recognition Using Correlation Filter Classifiers", *Proceedings of 38th Asilomar Conference on Signals, Systems and Computers*, Vol. 1, p. 567571, 2004.

79. Poon, C., D. C. M. Wong, and H. C. Shen, "A New Method in Locating and Segmenting Palmprint into Region-of-Interest", *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 4, pp. 533–536, 2004.

80. You, J., A. W.-K. Kong, D. Zhang, and K. H. Cheung, "On Hierarchical Palmprint Coding with Multiple Features for Personal Identification in Large Databases", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 2, pp. 234–243, 2004.

81. Li, W., J. You, and D. Zhang, "Texture-Based Palmprint Retrieval Using a Layered Search scheme for personal identification", *IEEE Transactions on Multimedia*, Vol. 7, No. 5, pp. 891–898, 2005.

82. Noh, J. S. and K. H. Rhee, "Palmprint Identification Algorithm Using Hu Invariant Moments and Otsu Binarization", *Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science*, pp. 94–99, 2005.

83. Xiong, W., K.-A. Toh, W.-Y. Yau, and X. Jiang, "Model-Guided Deformable Hand Shape Recognition without Positioning Aids", *Pattern Recognition*, Vol. 38, No. 10, pp. 1651–1664, 2005.

84. Su, C.-L., "Original Finger Image Extraction by Morphological Technique and Finger Image Comparisons for Persons' Identification", *Journal of Intelligent and Robotic Systems*, Vol. 45, No. 1, pp. 1–14, 2006.

85. Keren, D., "Using Symbolic Computation to Find Algebraic Invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 11, pp. 1143–1149, 1994.

86. Fouquier, G., L. Likforman, J. Darbon, and B. Sankur, "The BIOSECURE Geometry-Based System for Hand Modality", *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, 2007.

87. Joshi, D. G., Y. V. Rao, S. Kar, V. Kumar, and R. Kumar, "Computer Vision-Based Approach to Personal Identification using finger crease pattern", *Pattern Recognition*, Vol. 31, No. 1, pp. 15–22, 1998.

88. Fratric, I. and S. Ribaric, "A Biometric Identification System Based on Eigenpalm and Eigenfinger Features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 11, pp. 1698–1709, 2005.

89. Ribaric, S. and I. Fratric, "An online biometric authentication system based on eigenfingers and finger-geometry", *Proceedings of the 13th European Signal Processing Conference*, 2005.

90. Wong, M., D. Zhang, W. Kong, and G. Lu, "Real-Time Palmprint Acquisition system design", *IEE Proceedings - Vision, Image and Signal Processing*, Vol. 152, No. 5, p. 527534, 2005.

91. Kumar, A. and D. Zhang, "Integrating Shape and Texture for Hand Verification", *Proceedings of the Third International Conference on Image and Graphics*, pp. 222–225, 2004.

92. Gokberk, B., H. Dutağacı, A. Ulas, L. Akarun, and B. Sankur, "Representation

plurality and fusion for 3-D face recognition", *IEEE Transactions on Systems Man and Cybernetics Part B*, Vol. 38, No. 1, pp. 155–173, February 2008.

93. Dutağacı, H., B. Gokberk, B. Sankur, and L. Akarun, "A Study on Region-Based Recognition of 3D Faces with Expression Variations", *Proceesings of the 15th European Signal Processing Conference*, 2007.

94. Chang, K., K. W. Bowyer, and P. J. Flynn, "Face Recognition Using 2D and 3D Facial Data", *ACM Workshop on Multimodal User Authentication*, pp. 25–32, 2003.

95. Lu, X., A. Jain, and D. Colbry, "Matching 2.5D Face Scans to 3D Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 1, pp. 31–43, 2006.

96. Papatheodorou, T. and D. Reuckert, "Evaluation of Automatic 4D Face Recognition Using Surface and Texture registration", *Sixth International Conference on Automated Face and Gesture Recognition*, pp. 321–326, 2004.

97. İrfanoğlu, M. O., B. Gokberk, and L. Akarun, "3D Shape based Face Recognition using Automatically Registered Facial Surfaces", *Proceedings of the International Conference on Pattern Recognition*, pp. 183–186, 2004.

98. Gokberk, B., A. A. Salah, and L. Akarun, "Rank-based Decision Fusion for 3D Shape-based Face Recognition", Kanade, T., A. Jain, and N. K. Ratha (editors), *Proceedings of Audio- and Video-based Biometric Person Authentication, Lecture Notes in Computer Science*, Vol. 3456, pp. 1019–1029, 2005.

99. Koudelka, M., M. Koch, and T. Russ, "A Prescreener for 3D Face Recognition Using Radial Symmetry and the Hausdorff Fraction", *IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

100. Passalis, G., I. Kakadiaris, T. Theoharis, G. Toderici, and N. Murtuza, "Evaluation of 3D Face Recognition in the Presence of Facial Expressions: An Annotated

Deformable Model Approach", *IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

101. Lu, X. and A. Jain, "Deformation Modeling for Robust 3D Face Matching", *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 30, No. 8, pp. 1346–1357, 2008.

102. Bronstein, A., M. Bronstein, and R. Kimmel, "Three-dimensional Face Recognition", *International Journal of Computer Vision*, Vol. 64, No. 1, pp. 5–30, 2005.

103. Mian, A., M. Bennamoun, and R. Owens, "Matching Tensors for Pose Invariant Automatic 3D Face Recognition", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 3, pp. 120–120, 2005.

104. Mian, A., M. Bennamoun, and R. Owens, "Face Recognition Using 2D and 3D Multimodal Local Features", *International Symposium on Visual Computing*, 2006.

105. Chang, K., K. Bowyer, and P. Flynn, "Adaptive Rigid Multi-Region Selection For Handling Expression Variation in 3D Face Recognition", *IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

106. Faltemier, T., K. Bowyer, and P. Flynn, "3D Face Recognition with Region Committee Voting", *Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission*, 2006.

107. Gokberk, B., M. O. Irfanoglu, L. Akarun, and E. Alpaydin, "Learning the Best Subset of Local Features for Face Recognition", *Pattern Recognition*, Vol. 40, No. 5, pp. 1520–1532, MAY 2007.

108. Gokberk, B. and L. Akarun, "Selection and Extraction of Patch Descriptors for 3D Face Recognition", *Proceedings of the Computer and Information Sciences*, Vol. 3733 of *Lecture Notes in Computer Science*, pp. 718–727, 2005.

109. Samir, C., A. Srivastava, and M. Daoudi, "Three Dimensional Face Recogni-

tion Using Shapes of Facial Curves", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 11, pp. 1858–1864, 2006.

110. Pan, G., S. Han, Z. Wu, and Y. Wang, "3D Face Recognition Using Mapped Depth Images", *IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

111. Russ, T., M. Koch, and C. Little, "A 2D Range Hausdorff Approach for 3D Face Recognition", *IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

112. Abate, A., M. Nappi, S. Ricciardi, and G. Sabatino, "Fast 3D Face Recognition Based On Normal Map", *IEEE International Conference on Image Processing*, Vol. 2, pp. 946–949, 2005.

113. Tanaka, H., M. Ikeda, and H. Chiaki, "Curvature-Based Face Surface Recognition Using Spherical Correlation Principal Directions for Curved Object Recognition", *Proceedings of the Third International Conference on Automated Face and Gesture Recognition*, pp. 372–377, 1998.

114. Lee, J. and E. Milios, "Matching Range Images of Human Faces", *International Conference of Computer Vision*, pp. 722–726, 1990.

115. Gordon, G., "Face Recognition Based on Depth and Curvature Features", *Computer Vision and Pattern Recognition*, 108-110, 1992.

116. Chua, C.-S., F. Han, and Y.-K. Ho, "3D Human Face Recognition Using Point Signature", *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 233–237, 2000.

117. Beumier, C. and M. Acheroy, "Automatic 3D Face Authentication", *Image and Vision Computing*, Vol. 18, No. 4, pp. 315–321, 2000.

118. Dutağacı, H., B. Sankur, and Y. Yemez, "3D Face Recognition by Projection-

Based Features", *Proceedings of SPIE Conference on Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents*, 2006.

119. Srivastava, A., X. Liu, and C. Hesher, "Face Recognition Using Optimal Linear Components of Range Images", *Image and Vision Computing*, Vol. 24, No. 3, pp. 291–299, 2006.

120. Dutağacı, H., B. Sankur, and Y. Yemez, "A comparison of data representation types, features types and fusion techniques for 3D face biometry", *Proceesings of the 14th European Signal Processing Conference*, 2006.

121. `http://www.sic.rma.ac.be/ beumier/DB/3d_rma.html`.

122. Gökberk, B., *Three-Dimensional Face Recognition*, Ph.D. Thesis, Bogazici University, 2006.

123. Santini, S. and R. Jain, "Beyond Query by Example", *Proceedings of the Sixth ACM International Conference on Multimedia*, pp. 345–350, 1998.

124. Mcinerney, T. and D. Terzopoulos, "Deformable Models in Medical Image Analysis: A Survey", *Medical Image Analysis*, Vol. 1, pp. 91–108, 1996.

125. Styner, M. and G. Gerig, "Medial Models Incorporating Shape Variability", *Proceedings of the 17th International Conference on Information Processing in Medical Imaging*, pp. 502–516, Springer, 2001.

126. Joshi, S. C., M. I. Miller, and U. Grenander, "On the Geometry and Shape of Brain Sub-Manifolds", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 11, pp. 1317–1343, 1997.

127. Godil, A., Y. Ressler, and P. Grother, "Face Recognition Using 3D Facial Shape and Color Map Information: Comparison and Combination", *In Proceedings of the SPIE*, pp. 351–361, 2006.

128. Vranic, D. V., *3D Model Retrieval*, Ph.D. Thesis, University of Leipzig, 2004.

129. Dutağacı, H., B. Sankur, and Y. Yemez, "Subspace Building for Retrieval of General 3D models", *Computer Vision and Image Understanding*, under review, 2008.

130. Paquet, E. and M. Rioux, "Nefertiti: A Query by Content Software for Three-Dimensional Models Databases Management", *First International Conference on Recent Advances in 3D Digital Imaging and Modeling*, Vol. 0, p. 345, 1997.

131. Bustos, B., D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić, "Feature-Based Similarity Search in 3D Object Databases", *ACM Computing Surveys*, Vol. 37, No. 4, pp. 345–387, 2005.

132. Tangelder, J. W. and R. C. Veltkamp, "A Survey of Content Based 3D Shape Retrieval Methods", *Proceedings of the Shape Modeling International*, pp. 145–156, IEEE Computer Society, 2004.

133. Iyer, N., S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani, "Three-Dimensional Shape Searching: State-of-the-Art Review and Future Trends", *Computer-Aided Design*, Vol. 37, No. 5, pp. 509–530, 2005.

134. Kazhdan, M., *Shape Representations and Algorithms for 3D Model Retrieval*, Ph.D. Thesis, Princeton University, 2004.

135. Akgül, C. B., *Density-Based Shape Descriptors and Similarity Learning for 3D Object Retrieval*, Ph.D. Thesis, Bogazici University, March 2007.

136. Goodall, S., *3-D Content-Based Retrieval and Classification with Applications to Museum Data*, Ph.D. Thesis, University of Southampton, March 2007.

137. Zaharia, T. and F. Preteux, "Three-Dimensional Shape-Based Retrieval within the MPEG-7 Framework", *Proceedings SPIE Conference on Nonlinear Image Processing and Pattern Analysis XII*, Vol. 4304, pp. 133–145, January 2001.

138. Osada, R., T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions", *ACM Transactions on Graphics*, Vol. 21, No. 4, pp. 807–832, October 2002.

139. Kriegel, H.-P., P. Kröger, Z. Mashael, M. Pfeifle, M. Pötke, and T. Seidl, "Effective Similarity Search on Voxelized CAD Objects", *Proceedings of the Eighth International Conference on Database Systems for Advanced Applications*, p. 27, 2003.

140. Kriegel, H.-P., S. Brecheisen, P. Kröger, M. Pfeifle, and M. Schubert, "Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects", *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 587–598, 2003.

141. Kazhdan, M., B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser, "A Reflective Symmetry Descriptor", *Proceedings of the 7th European Conference on Computer Vision*, pp. 642–656, May 2002.

142. Funkhouser, T., P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A Search Engine for 3D Models", *ACM Transactions on Graphics*, Vol. 22, No. 1, pp. 83–105, January 2003.

143. Ricard, J., D. Coeurjolly, and A. Baskurt, "Generalizations of Angular Radial Transform for 2D and 3D Shape Retrieval", *Pattern Recognition Letters*, Vol. 26, No. 14, pp. 2174–2186, October 2005.

144. Novotni, M. and R. Klein, "A Geometric Approach to 3D Object Comparison", *Proceedings of the International Conference on Shape Modeling and Applications*, p. 167, 2001.

145. Novotni, M. and R. Klein, "3D Zernike Descriptors for Content Based Shape Retrieval", *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, pp. 216–225, 2003.

146. Suzuki, M. T., T. Kato, and N. Otsu, "A Similarity Retrieval of 3D Polygonal

Models Using Rotation Invariant Shape Descriptors", *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, pp. 2946–2952, 2000.

147. Sánchez-Cruz, H. and E. Bribiesca, "A Method of Optimum Transformation of 3D Objects Used as a Measure of Shape Dissimilarity", *Image and Vision Computing*, Vol. 21, No. 12, pp. 1027–1036, 2003.

148. Dutağacı, H., B. Sankur, and Y. Yemez, "Transform-Based Methods for Indexing and Retrieval of 3D Objects", *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pp. 188–195, 2005.

149. Kazhdan, M., B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A Reflective Symmetry Descriptor for 3D Models", *Algorithmica*, Vol. 38, No. 1, pp. 201–225, 2003.

150. Gal, R. and D. Cohen-Or, "Salient Geometric Features for Partial Shape Matching and Similarity", *ACM Transactions on Graphics*, Vol. 25, No. 1, pp. 130–150, 2006.

151. Paquet, E., "Description of Shape Information for 2-D and 3-D Objects", *Image Communication*, Vol. 16, pp. 103–122, September 2000.

152. Akgül, C. B., B. Sankur, Y. Yemez, and F. Schmitt, "Density-Based 3D Shape Descriptors", *EURASIP Journal of Applied Signal Processing*, Vol. 2007, No. 1, pp. 209–209, 2007.

153. Akgül, C. B., B. Sankur, Y. Yemez, and F. Schmitt, "3D Model Retrieval using Probability Density-Based Shape Descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in print, 2009.

154. Papadakis, P., I. Pratikakis, S. Perantonis, and T. Theoharis, "Efficient 3D Shape Matching and Retrieval Using a Concrete Radialized Spherical Projection Representation", *Pattern Recognition*, Vol. 40, No. 9, pp. 2437–2452, 2007.

155. Vranic, D. V. and D. Saupe, "3D Model Retrieval", *Spring Conference on Computer Graphics and its Applications*, pp. 89–93, 2000.

156. Vranic, D. V., D. Saupe, and J. Richter, "Tools for 3D-Object Retrieval: Karhunen-Loeve Transform and Spherical Harmonics", *Proceedings of IEEE Workshop on Multimedia Signal Processing*, pp. 293–298, 2001.

157. Ricard, J., D. Coeurjolly, and A. Baskurt, "ART Extension for Description, Indexing and Retrieval of 3D Objects", *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 3, pp. 79–82, 2004.

158. Vranic, D. V. and D. Saupe, "3D Shape Descriptor Based on 3D Fourier Transform", *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services*, pp. 271–274, Budapest, Hungary, September 2001.

159. Vranic, D. V., "An Improvement of Rotation Invariant 3D Shape Descriptor Based on Functions on Concentric Spheres", *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, pp. 757–760, 2003.

160. Liu, Y., J. Pu, H. Zha, W. Liu, and Y. Uehara, "Thickness Histogram and Statistical Harmonic Representation for 3D Model Retrieval", *Proceedings of the Second International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 896–903, 2004.

161. Kazhdan, M., T. Funkhouser, and S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors", *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 156–164, 2003.

162. Bowyer, K. W., K. Chang, and P. Flynn, "A Survey of Approaches and Challenges in 3D and Multi-Modal 3D + 2D Face Recognition", *Computer Vision and Image Understanding*, Vol. 101, No. 1, pp. 1–15, 2006.

163. Azouz, Z. B., C. Shu, R. Lepage, and M. Rioux, "Extracting Main Modes of Human Body Shape Variation from 3-D Anthropometric Data", *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pp. 335–342, 2005.

164. Ruto, A., M. Lee, and B. Buxton, "Comparing Principal and Independent Modes of Variation in 3D Human Torso Shape Using PCA and ICA", *ICA Research Network International Workshop*, 2006.

165. Kazhdan, M., "An Approximate and Efficient Method for Optimal Rotation Alignment of 3D Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 7, pp. 1221–1229, 2007.

166. Aggarwal, C. C., A. Hinneburg, and D. A. Keim, "On the Surprising Behavior of Distance Metrics in High Dimensional Space", *Lecture Notes in Computer Science*, pp. 420–434, Springer, 2001.

167. Noy-Meir, I., "Data Transformations in Ecological Ordination: I. Some Advantages of Non-Centering", *The Journal of Ecology*, Vol. 61, No. 2, pp. 329–341, 1973.

168. Shilane, P., P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark", *Proceedings of the IEEE International Conference on Shape Modeling and Applications*, pp. 167–178, 2004.

169. Chen, D.-Y. and M. Ouhyoung, "A 3D Object Retrieval System Based on Multi-Resolution Reeb Graph", *Proceedings of Computer Graphics Workshop*, 2002.

170. Zass, R. and A. Shashua, "Nonnegative Sparse PCA", Schölkopf, B., J. Platt, and T. Hoffman (editors), *Advances in Neural Information Processing Systems 19*, pp. 1561–1568, MIT Press, Cambridge, MA, 2007.

171. Huang, K. and S. Aviyente, "Sparse Representation for Signal Classification",

NIPS, pp. 609–616, 2006.

172. Torre, F. D. L. and M. J. Black, "A Framework for Robust Subspace Learning", *International Journal on Computer Vision*, Vol. 54, No. 1-3, pp. 117–142, 2003.

173. Spratling, M. W., "Learning Image Components for Object Recognition", *Journal of Machine Learning Research*, Vol. 7, pp. 793–815, 2006.

174. Soukup, D. and I. Bajla, "Robust Object Recognition under Partial Occlusions Using NMF", *Computational Intelligence and Neuroscience*, 2008.