ROBUST GEOMETRICALLY INVARIANT FEATURES FOR 2D SHAPE MATCHING AND 3D FACE RECOGNITION

by

Wei-Yang Lin

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2006

To my wife Chia-Ling , and our son Daniel.

ACKNOWLEDGMENTS

I have been privileged to have Professor Nigel Boston and Professor Yu Hen Hu as my advisers. Their altitude toward life and academic research has inspired me a lot. I would like to express my deep gratitude to them for their support through the years, and their invaluable guidance on my PhD study. Professor Charles Dyer has long been an inspiration to me. Thanks to many fruitful discussion with him during my Ph.D. study.

Thanks to the members in the committee of my preliminary exam, Professor John Gubner and Professor Robert Nowak, for their suggestions and critiques on the earlier works. Their feedbacks have proved to be very important to this dissertation.

I would also like to thank the members in the face recognition group. They are Chau Ha ”Jenice” Fung, Zhaozheng Yin, Kin-Chung ”Ryan” Wong, Harris Nover, Matt Darnall, Guodong Guo, Jay Wierer, Kerry Widder, Tony Boehm, and Adam Slater. The weekly meeting is one of my most productive learning experiences at UW-Madison. Special thanks to Kin-Chung ”Ryan” Wong who has collaborated with me on the FRGC experiments. My appreciation to Kerry Widder for proofreading my dissertation.

Thanks to Dr. I-Chang Jou in Taiwan for encouraging me to pursue a career in academic research.

I would like to take this special occasion to thank my parents and parents-in-law back in Taiwan for letting me pursue my dream. Finally, I would like to thank my wife Chia-Ling Su for her incredible support, encouragement and love.

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES vii

ABSTRACT x

1 Introduction 1

1.1 Geometric Transformations 3
1.2 History of Geometric Invariants 5
1.2.1 Algebraic Invariants 6
1.2.2 Moment Invariants 6
1.2.3 Fourier Descriptor 7
1.2.4 Wavelet Invariants 8
1.3 Motivation 9
1.4 Objectives and Contributions of Dissertation 11
1.5 Dissertation Organization 12

2 Cartan's Moving Frame Method and Its Application to Object Recognition 14

2.1 Background 14
2.1.1 Lie Groups 14
2.1.2 Orbits and Invariant Functions 15
2.1.3 Jet Space and Prolongations 16
2.1.4 Method of Moving Frames 17
2.2 Related Works 20
2.2.1 Approximation of Differential Invariants 20
2.2.2 Invariant Parameterization 21
2.2.3 Integral Invariant Signatures 22
2.2.4 Integral Invariant 22
2.3 Summation Invariants 24
2.3.1 Summation Invariants for Curves in $\mathbb{R}^2$ 25
2.3.2 Summation Invariants for Surfaces in $\mathbb{R}^3$ 30
2.3.3 A Numerical Example 36
2.4 Robustness in the Presence of Noise 38
2.4.1 Error Analysis on Summation Invariants 39
2.4.2 Error Analysis on Integral Invariants 42

Page

2.4.3 Experiment Results . . . 44
2.5 Comparison with Curvature . . . 46
2.6 Summary . . . 50

3 2D Shape Matching Under Visual Deformation . . . 53

3.1 Introduction . . . 53
3.2 Representation of Curves using Summation Invariants . . . 54
3.3 Shape Matching Experiments Using Synthesized Data . . . 56
3.3.1 Generating the Shape Dataset . . . 57
3.3.2 Experimental Setup and Results . . . 60
3.3.3 Sensitivity to Noise . . . 63
3.4 Nonrigid Matching Using Summation Invariants . . . 64
3.4.1 Symmetric Problem Formulation . . . 65
3.4.2 Finding the Optimal Alignment Curve by Dynamic Programming . . . 69
3.4.3 Experimental Results . . . 73
3.5 Summary . . . 74

4 3D Face Recognition . . . 78

4.1 Introduction . . . 78
4.1.1 Technical Terminology . . . 79
4.1.2 Representation Techniques . . . 80
4.1.3 Classification Techniques . . . 80
4.1.4 Previous Work in 3D Face Recognition . . . 82
4.2 Face Databases . . . 86
4.3 3D Face Recognition Using Summation Invariants . . . 88
4.3.1 3D Face Dataset and BEE . . . 89
4.3.2 Single Region Face Recognition Algorithm . . . 89
4.3.3 Experimental Results . . . 92
4.4 Summary . . . 99

5 Optimal Linear Fusion for Improving Performance . . . 101

5.1 Literature Review . . . 102
5.2 Multi-Regions 3D Face Recognition Algorithm . . . 104
5.2.1 Description of Experiments . . . 105
5.2.2 Sum Rule . . . 106
5.2.3 Linear Discriminant Analysis . . . 106
5.2.4 Linear Support Vector Machine . . . 109

Page

5.2.5 Nonlinear Fusion 112
5.2.6 Discussion 114
5.2.7 Sensitivity to Facial Expressions 119
5.3 Summary 121

6 Conclusion and Future Works 123

6.1 Summary of the Work 123
6.2 Future works 125

LIST OF REFERENCES 128

LIST OF TABLES

Table Page

3.1 Error Rate of Fish Recognition at Different Noise Level 63

4.1 Representation Techniques and Their Applications to Face Recognition 81

4.2 3D Face Recognition Algorithms Using Invariant Features 85

4.3 Comparison of 2D Face Databases 88

5.1 The Weights on Selected Regions Obtained by LDA 110

5.2 The Weights on Selected Regions Obtained by LSVM 113

5.3 The Verification Rates of Fusion by Polynomial SVM 116

5.4 The Verification Rates of Fusion by RBF SVM 116

LIST OF FIGURES

Figure Page

2.1 Method of moving frame: original curve (blue) and the curve transformed by moving frame (red). 20

2.2 Two surfaces related by Euclidean transformation 37

2.3 The variance of quantization error $\Delta\nu_{0,1}$ 46

2.4 The variance of quantization error $\Delta\eta_{0,1}$ 47

2.5 The variance of relative error $\frac{\Delta\eta_{0,1}}{\eta_{0,1}}$ and $\frac{\Delta\nu_{0,1}}{\nu_{0,1}}$ 48

3.1 (a) and (b) are two fish contours related by affine transformation. Their semi-local summation invariants $\lambda_{affine}[m]$ are shown in (c) and (d) respectively. Note (c) and (d) are related by translation because the initial point on the fish contour is randomly chosen. 56

3.2 Database contains 100 prototype fish contours. Each prototype represents one distinct type of fish. 59

3.3 Variations of fish contours for type 1 in database. Contour deformations are generated by applying affine transformations on a fish prototype. Twenty deformations are produced for each prototype. 60

3.4 Fish contours before and after adding Gaussian distributed noise.(σ denotes the standard deviation of noise) 63

3.5 By introducing the alignment curve α in the problem formulation, the optimal solution allows for a segment from one curve corresponding to a single point on the other curve, *i.e.* many-to-one correspondence, as well as a single point from one curve corresponding to a segment in the other curve, *i.e.* one-to-many correspondence. . . . 66

3.6 An infinitesimal curve segment with arc-length ds. The tangent vectors at point A and point B are denoted by T_A and T_B respectively. 70

Figure Page

3.7 An example of DP table. Discrete sample points along C and $\tilde{C}$ are the x-axis and y-axis respectively. The value of $d(i,j)$, stored at the entry (i,j), is updated by considering the 9 possible paths shown here. 71

3.8 4 cases to be considered during shape matching. (a) The original shape, (b) The symmetric shape, (c) The original shape with opposite starting point, and (c) The symmetric shape with opposite starting point. The starting point on each curve is denoted by a red dot. 72

3.9 An open query and its 20 best matches retrieved by the proposed shape matching algorithm. 74

3.10 An open query and its 20 best matches retrieved by Petrakis *et. al.*'s algorithm [1]. . . 75

3.11 An closed query and its 20 best matches retrieved by the proposed shape matching algorithm. 76

3.12 An closed query and its 20 best matches retrieved by Petrakis *et. al.*'s algorithm [1]. . 77

4.1 (a) The computed result of $\eta_{0,1}$ using a horizontal window; and (b) The computed result of $\eta_{0,1}$ using a vertical window. 91

4.2 Normalized depth map and an 81×81 region centered at the nose tip. 92

4.3 ROC performance for using different unit difference on the x and y coordinates. The results indicate that geometrical distortion will cause performance degradation of summation invariants. (a) ROC curves of $\eta_{0,1}$ and (b) ROC curves of $\kappa_{0,0,1}$. 94

4.4 ROC performance obtained using different summation invariants. (a) $\eta_{i,j}$ and (b) $\kappa_{i,j,k}$ 95

4.5 ROC performance obtained by combining two summation invariants. (a) Fusion of $\eta_{1,1}$ and others. (b) fusion of $\kappa_{0,0,1}$ and others. 97

4.6 Comparison with FRGC 3D baseline algorithm. We apply the FRGC 3D baseline algorithm on the normalized depth map and the cropped region shown in Fig 4.2. Their corresponding ROC curves are shown by the solid line and the dash line respectively. . 98

4.7 Failure example of the FRGC 3D normalization process: (a) Original range image and (b) Normalized range image. 100

Figure Page

5.1 Example of opinion/score fusion. The scores from mouth region and nose are combined to yield a final score. 103

5.2 We specify 10 regions on facial surface. Matching scores obtained from each region are combined to yield the final matching score. 105

5.3 ROC performance obtained by using sum rule for information fusion. (a) Comparison with BEE baseline algorithm and (b) Comparison with single-region algorithm 107

5.4 ROC performance obtained by using LDA for information fusion. 109

5.5 ROC performance obtained by using LSVM for information fusion. 112

5.6 Nonlinear fusions of region #1 (*horizontal axis*) and region #5 (*vertical axis*). Decision boundaries (*black lines*) and marginal boundaries (*green and blue lines*) are created by using training partition of FRGC v2.0 dataset. (a) Polynomial SVM, (b) RBF SVM. 115

5.7 Score distribution of region 1,2,3,4,5,6. Higher score means more similar. (dashed lines denote the Gaussian approximation of score distribution). 118

5.8 Score distribution of region 7,8,9,10. Higher score means more similar. (dashed lines denote the Gaussian approximation of score distribution). 119

5.9 Score distribution of combining 10 regions. (a) Sum rule. (b) LDA. 120

5.10 ROC performance for different levels of expression strength. 122

ABSTRACT

Invariant features play a key role in object and pattern recognition studies. Features that are invariant to geometrical transformations offer succinct representations of underlying objects so that they can be reliably identified.

In this dissertation, we introduce a family of novel invariant features based on Cartan's theory of moving frames. We call these new features summation invariants. Compared to existing invariant features, summation invariants are inherently numerically stable, and do not require computationally complex numerical integrations or analytical representations of underlying data. We develop robust methods for extracting summation invariants from sampled 2D contours and 3D surfaces. We further apply these new invariant features to 2D and 3D object recognition problems.

In an application to a 2D shape recognition problem, we compare the performance of the proposed 2D contour summation invariant features with that of integral invariant features as well as wavelet invariant features. We observe marked performance enhancement achieved by the new summation invariant features.

The summation invariant features are also successfully applied to 3D face recognition applications. We propose robust methods to extract summation invariant features based on 2D contours and 3D shapes of given facial range images. We further work out an optimal feature selection and decision fusion method to select the most discriminating invariant features. The same method also facilitates the development of a multi-region face recognition method that achieves higher performance than using a monolithic facial image. To validate the proposed novel 3D face recognition algorithms, we test them on the Face Recognition Grand Challenge (FRGC) version 2 dataset with

a data corpus of more than 50,000 facial images. The multi-region summation-invariant algorithm outperforms the best results in the recent FRGC report.

To conclude, we introduce a systematic approach for constructing robust geometrically invariant features. The proposed features provide improved accuracy and are applicable to a wide range of pattern recognition applications. These are versatile features that can be adapted to an engineer’s choice of transformation group, such as rigid, affine, or similarity transformation group, to mention a few.

Chapter 1

Introduction

Object recognition is a major goal of many computer vision applications, but many unsolved issues characterize the development of practical recognition systems. A typical object recognition problem can be described as below. An object can be observed from a different point of view, and hence, the appearance of the same object will vary. For the objects we want to recognize, their templates, called *gallery images*, are stored in a database. The task is to match an unknown object observed from an unknown point of view, called the *probe image*, against all the templates in a database.

Now, let us discuss a little bit how an object recognition system works. Given an image of an object, one wants to use an algorithm that produces a unique feature from the image. Given another image of the same object, the algorithm should produce exactly the same feature. To achieve this, the algorithm has to cancel the effect of shape transformations which is the result of changing viewing angles.

There are many approaches by which one can deal with shape transformations between images. An intuitive approach is simply to perform every possible transformation of a template to see whether any of the template's transformed versions match the probe image. For example, if a probe image and a template differ only by translation [2], then one can shift the template, pixel by pixel, until the best match is reached. Obviously, this kind of method is computationally intensive.

Also, for common transformations in computer vision, such as affine transformation and projective transformation, the search space becomes overwhelmingly large.

Another method is to store images of an object from all possible viewing angles in the database. By doing this, one can reduce the computational complexity significantly, but at the price of increasing the storage requirement. Once again, for the common transformations in computer vision, this approach is not an efficient solution and probably not feasible.

Due to the inherent drawbacks of the previous two approaches, it is necessary to come up with a more practical solution for real world applications. One can use the concept of geometric invariance to tackle this challenging problem. Ideally, the invariant descriptors are independent of viewpoints. By using this kind of shape descriptor, one can match two images directly without any prior knowledge of the position and the orientation of the camera. Therefore, the invariant-based method is neither expensive on computation nor storage.

For an invariant descriptor, another important property is that it should extract all the distinctive information about a shape. Thus, one can identify different shapes by using an invariant feature. However, an invariant feature does not necessarily have good discrimination ability. For example, a constant function provides perfect invariance, that is, it will yield the same value for a shape under arbitrary transformations. The problem is that the constant function will also produce the same number for different shapes. In other words, one cannot use the constant function to distinguish two different shapes. This might be an extreme example, but it points out a fundamental issue associated with the invariant-based approach. Basically, the more invariance properties an invariant feature has, the less discrimination capability it can provide. In other words, one has to make a trade off between invariance property and discrimination ability. Of course, how to choose a good trade off is highly application-specific.

The trade off consideration leads to the following conclusion: the optimal invariant descriptors are the properties of shapes that will stay unchanged only under a particular class of transformations, not under a larger class of transformations. The paradigm of invariant-based object recognition can be described as follows:

1. Identify the transformations that an object can undergo in a particular application. In other words, find a mathematical formulation that can model all the possible shape deformations in a particular application.
2. Find the descriptors that are invariant to these transformations but not to others.
3. Use these descriptors as shape representations for matching.

In the following sections, we will discuss both common transformations in computer vision and approaches by which one can find invariant descriptors.

1.1 Geometric Transformations

In this dissertation, we will focus on geometric properties, that is, properties *intrinsic* to the shape itself. The treatment of *extrinsic* properties, like shading, reflectance, and color, that can function as invariant descriptors, is beyond the scope of my dissertation.

The invariants to the Euclidean transformations, *i.e.* combinations of rotation and translation, frequently surface in vision problems. For example, the length of a rod is a Euclidean invariant. Consider a very simple scenario where a fixed table is parallel with the image plane. Suppose that a rod on the table does not have a fixed location and orientation: one can easily identify a particular rod by measuring its length in an image. As another example, when a surface is rotated and translated in 3D space, its principal curvatures at each point do not change. Thus, the principal curvatures are Euclidean invariants.

Imaging processing is a larger class of transformations called *projective transformations*. When enlarging the set of transformations, the invariants of a smaller set do not remain invariants. For example, the length of a rod in an image will change according to different points of view. Similarly, curvatures are not preserved under such transformations. The degree of freedom in this case is eight, so finding projective invariants is not a trivial task as it is in the Euclidean case. In order to keep the analysis as simple as possible, one should not enlarge the set of transformations beyond what is necessary. The trade off consideration, mentioned before, leads to the same conclusion.

In fact, one cannot use purely geometrical methods to find the invariants of projection from 3D space to image plane. The reason is that all the points along a projection ray are mapped to a point on an image plane; thus, the true depth information is unrecoverable. *Model-based* technologies can facilitate the interpolation of missing data, but this fact is not the main focus of my dissertation. It is common to use subsets of the projective transformations to simplify the task of finding invariants. For example, one can consider the projection of planar objects and find the corresponding invariants. In many vision applications, 3D objects contain planar shapes, such as facets. Hence, the 2D projective transformations and their invariants can facilitate one's recognition of many 3D objects.

When objects are relatively small, one can treat projection rays as nearly parallel. On the basis of this assumption, one can use affine transformations that have six degrees of freedom to approximate the projective transformations. The affine invariants are often quite useful in 3D applications.

In sum, specifying a set of transformations is of crucial importance in invariant-based object recognition. The set of transformations should no larger than what is necessary for the following two reasons:

1. Invariance that is too general will decrease the discrimination capability of an invariant feature.

2. It is more difficult to find invariants for a larger set of transformations.

For some transformations, finding their invariants is difficult or impossible. Under these kinds of scenarios, one could simplify the problem by reducing the size of transformations. This kind of simplification has proven useful in many vision applications.

1.2 History of Geometric Invariants

In the field of computer vision, the importance of invariant descriptors has been recognized since its origin in the 1960s. The emphasis was initially on *photometric invariants*, and this led to the development of edge detectors. The early 1980s witnessed the emergence of an interest in the theory of *geometric invariants* (invariants that are geometric properties remaining unchanged under a particular class of transformations).

In computer vision, the primary interest is in mapping between 3D space and 2D image planes. Historically, most research in vision problems centered on mapping between planes, that is, mapping from a plane in 3D space to an image plane. One of the main reasons for this historical focus is that when spaces are of the same dimension, the transformations often form a group. Group theory applies mathematical machinery to the construction of invariants. It is therefore not surprising that most invariants in computer vision applications are based on plane-to-plane mappings.

From the perspective of group theory, progress was made in finding invariants of differentiable manifolds such as curves and surfaces. These differential invariants are local features because they are measured at each point of a shape. Among the methods that one can use to obtain such features, there is one that clearly stands out owing to its simplicity and generality. This method is known as Cartan's moving frame method and was introduced by Élie Cartan [3] in the middle of the 1930s. The method is widely used in mathematics and physics, but less well known to the computer vision community. In computer vision, only very limited kinds of differential invariants are used. The curvatures, which are Euclidean invariants, are most frequently used. Other differential invariants obtained by the *method of moving frames* have been applied in computer vision problems [4, 5]. One can systematically generate the invariants for different kinds of transformations by using the method of moving frames. Theoretically, one can produce as many different invariants as one wants, and the resulting differential invariants are functionally independent.

Many other research works on the pursuit of geometric invariants have appeared in the literature. In the following sections, we will briefly discuss some popular geometric invariants in the area of computer vision.

1.2.1 Algebraic Invariants

Algebraic invariants have successfully facilitated efforts to recognize industrial objects [6]. One can obtain these invariants by fitting polynomials to the shape and then by calculating invariants from the polynomial coefficients. The algebraic approach is problematic in several regards. First, most shapes cannot be expressed in terms of simple polynomials, and this makes the curve fitting difficult. Second, algebraic invariants are a global method because they require the whole shape, preventing this approach from working under partial occlusion.

1.2.2 Moment Invariants

The theory of moment invariants for object recognition was first introduced by Hu in 1962 [7]. Given the intensity function of an image $f(x, y)$, one can define moment m_{pq} as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy, \; p, q = 0, 1, 2, \ldots . \tag{1.1}$$

Moment invariants of 2D images are built on the basis of the moments defined above. Summation invariants, to be defined later in this dissertation, and moment invariants are similar in the sense that they all depend on summation computations. On the other hand, they are different in many other regards. Moment is defined globally, that is, the moment requires the whole shape. Hence, moment invariants are known to suffer from occlusion problems. In [8], four absolute invariants of general linear transformations are derived. The resulting invariant feature of an image is a 4×1 vector. The feature space of moment invariants is quite small compared with that of summation invariants, as we shall see.

1.2.3 Fourier Descriptor

In [9], affine-invariant Fourier descriptors are used in airplane recognition. The authors construct the affine-invariant Fourier descriptors by eliminating dependencies on the affine transformation in the Fourier domain. They assume that the contours are closed and without self-intersection.

Consider a parameterized curve,

$$x(t) = \begin{bmatrix} u(t) \\ v(t) \end{bmatrix} \tag{1.2}$$

Fourier transform is then applied to $u(t)$ and $v(t)$, resulting in Fourier coefficients U and V. $U(0)$, $V(0)$ are ignored since they depend on translation. The remaining coefficients are all translation invariants. Affine-invariant Fourier descriptors are derived as follows: Let $\overline{X}_k$ represent the k^{th} Fourier coefficient of the affine transformed contour. Similarly, X_k represents the same coefficient from the Fourier transform of original contour. Since Fourier transform is a linear operation, one has

$$\overline{X}_k = \mathbf{A}X_k \tag{1.3}$$

,where $\mathbf{A}$ is a 2×2 matrix with real entries. Choosing any two coefficients, say k and p, and constructing 2×2 matrix

$$\begin{bmatrix} X_k & X_p \end{bmatrix} \tag{1.4}$$

one can show that

$$\begin{bmatrix} \overline{X}_k & \overline{X}_p \end{bmatrix} = \mathbf{A} \begin{bmatrix} X_k & X_p \end{bmatrix} \tag{1.5}$$

Taking the determinant of both sides, one obtains a relative affine invariant as below:

$$\det \begin{bmatrix} \overline{X}_k & \overline{X}_p \end{bmatrix} = \det \mathbf{A} \det \begin{bmatrix} X_k & X_p \end{bmatrix} \tag{1.6}$$

This method cannot extract local characteristics in the space domain because a Fourier transformation requires the whole contour. One can control sensitivity to noise by ignoring detail coefficients. The number of dropped coefficients is experimentally determined. This approach is similar to how the window of semi-local summation invariants is determined. Because one is free to use either a coefficient or its complex conjugate in Equation (1.6), one has $16N^2$ invariants for N coefficients. The resulting feature space is much larger than that of moment invariants.

1.2.4 Wavelet Invariants

Researchers have developed several techniques to preserve locality, that is, to restrict the computation to a local neighborhood: wavelet transformation [10] and potential [11]. In this section, we will discuss wavelet-based invariants in detail.

Wavelet transformation is a way to represent a signal in multiple resolution levels. This transformation generates approximation and detail coefficients. Let $W_i x[n]$ denote the detail coefficients of $x[n]$ at resolution level i. Then, the affine transformation of x[n] can be written as below:

$$W_i \overline{x}[n] = aW_i x[n] + bW_i y[n] \tag{1.7}$$

Note that the wavelet transformation of a constant signal will be always zero. A very simple affine invariant function involving two resolution levels can be defined as:

$$\eta_{i,j} = W_i x[n] \cdot W_j y[n] - W_i y[n] \cdot W_j x[n], i \neq j \tag{1.8}$$

But, different shape might have the same representation at some resolution levels. In order to improve the discriminating power, one trivial solution is to incorporate more resolution levels in one affine invariant function. Khalil *et al.* [10] proposed a wavelet affine invariant

$$
\begin{aligned}
\eta_{a,b,c,d,e,f}[n] = & \begin{vmatrix} 1 & 2W_ax[n]W_ay[n] & W_a^2y[n] \\ 1 & 2W_bx[n]W_by[n] & W_b^2y[n] \\ 1 & 2W_cx[n]W_cy[n] & W_c^2y[n] \end{vmatrix} \begin{vmatrix} W_d^2x[n] & 2W_dx[n]W_dy[n] & 1 \\ W_e^2x[n] & 2W_ex[n]W_ey[n] & 1 \\ W_f^2x[n] & 2W_fx[n]W_fy[n] & 1 \end{vmatrix} \\
& + \begin{vmatrix} W_a^2x[n] & 2W_ax[n]W_ay[n] & 1 \\ W_b^2x[n] & 2W_bx[n]W_by[n] & 1 \\ W_c^2x[n] & 2W_cx[n]W_cy[n] & 1 \end{vmatrix} \begin{vmatrix} 1 & 2W_dx[n]W_dy[n] & W_d^2y[n] \\ 1 & 2W_ex[n]W_ey[n] & W_e^2y[n] \\ 1 & 2W_fx[n]W_fy[n] & W_f^2y[n] \end{vmatrix} \\
& - 2\begin{vmatrix} W_a^2x[n] & 1 & W_a^2y[n] \\ W_b^2x[n] & 1 & W_b^2y[n] \\ W_c^2x[n] & 1 & W_c^2y[n] \end{vmatrix} \begin{vmatrix} W_d^2x[n] & 1 & W_d^2y[n] \\ W_e^2x[n] & 1 & W_e^2y[n] \\ W_f^2x[n] & 1 & W_f^2y[n] \end{vmatrix}
\end{aligned}
$$

, where $\{a, b, c, d, e, f\}$ are indices to six different resolution levels. One can control wavelet invariants' sensitivity to noise by choosing an appropriate resolution level. Wavelet invariants usually ignore the finer resolution levels because they are sensitive to noise. In practical applications, it is necessary to also perform re-parameterization so that contours have the same length. The computation complexity of dyadic wavelet transformation is $O(N \log N)$, where N is the number of sampling points on a contour. The semi-local summation invariants have lower computation complexity, which is $O(N)$.

1.3 Motivation

The moving frame method is completely general and can be used in many cases. However, the numerical computation of differential invariants is neither accurate nor reliable. The main reason is that they depend on high-order derivatives and, thus, are particularly sensitive to noise and round-off error. To reduce the noise-induced fluctuation, researchers have proposed *semi-differential invariants* that depend on lower-order derivatives [12, 13]. The related works include *noise-resistant* differential invariants developed by Weiss [14], *local invariant signature* by Bruckstein *et al.* [15] and *numerically invariant signature* by Calabi *et al.* [4].

In order to overcome the limitations of differential invariants, researchers have tried to derive invariants based on integrals. A statistical approach using moments was introduced in [7]. Moment invariants under affine transformations were derived in [16]. The moment invariants are global features, and therefore, the resulting feature space has low dimensionality. It is difficult to distinguish similar shapes in such a low-dimensional feature space. The error analysis and the analytic characterization of moment descriptors were studied in [17]. Another research project used Fourier transformation to obtain invariants [9]. One can represent a closed curve by using a set of Fourier coefficients and then use normalized Fourier descriptors to obtain affine invariants. In general, this method features high-order Fourier coefficients that are sensitive to noise. Sato and Cipolla formulated semi-local integral invariants by using the affine quasi-invariant arc-length [18]. Hann and Hickman used potentials to prolong group actions, resulting in invariants consisting of integrals rather than derivatives [11].

From my point of view, a desirable invariant feature should have the following properties.

1. **Systematically generated:** The algorithm should be applicable to different transformation groups and produces corresponding invariant features.

2. **Multiple invariants:** It is highly desirable to have multiple invariants for a particular transformation group. A combination of invariants will provide performance improvement over any individual invariant.

3. **Robustness:** Because sensor noise and quantization in the data are inevitable, an invariant feature should exhibit immunity to these perturbations.

These properties serve as a guideline for the design of new invariant features. In this dissertation, we propose a novel scheme, called *summation invariants*, by which one can construct invariants whose description are available through summation operations. We explicitly derive summation invariants for both curves on a plane and surfaces in 3D space. The proposed scheme can be considered a generalization of the work by Hann and Hickman [11]. One major difference is that their approach is not generalizable to 3D cases. Here, the groups that are of interest to me are the Euclidean and the affine groups acting on $\mathbb{R}^2$ and $\mathbb{R}^3$.

1.4 Objectives and Contributions of Dissertation

In this dissertation, we propose a frame work for summation invariants along with four important classes of summation invariants. The proposed invariant features are robust in the presence of noise and quantization error. For different transformation groups and different objects, their corresponding summation invariants are generated by the same procedure. The summation invariants for a particular transformation group are functionally independent, and therefore, one can obtain performance improvement by combining multiple summation invariants. We use the summation invariants to define a novel shape representation which is useful across many applications. Two major applications are listed below.

1. **2D shape recognition:** We derive the summation invariants of curves under both Euclidean transformation and affine transformation. We also demonstrate the specific application of the matching problem of 2D shapes under nonrigid transformations. The experimental results show that summation invariants provide better discrimination capability and noise immunity than do some conventional invariant features.

2. **3D shape recognition:** We derive the summation invariants of surfaces under both Euclidean transformation and affine transformation. We propose a 3D face recognition algorithm based on summation invariants. Experimental results concerning the Face Recognition

Grand Challenge (FRGC) dataset show significant performance improvement by using summation invariants. We observe a further improvement by combining multiple summation invariants.

In the following chapters, we present a detailed description of each of these contributions.

1.5 Dissertation Organization

The following information concerns the organization of this dissertation:

In chapter 2, we present an introduction on a modern treatment of the theory of geometric invariants. Specifically, we discuss the mathematical definitions of *Lie group*, *orbit*, *invariant function*, *jet space*, *group-action prolongation*, and *the method of moving frames*. This discussion will lead to a problem formulation and to definition of terminologies. In this chapter, we also present various differential and integral invariants that are known in the literature. We then discuss, case by case, summation invariants for a specific transformation group. Also, we present a comparison of integral invariants and summation invariants.

In chapter 3, we review some existing invariants that have been used frequently by researchers for 2D shape recognition. In this regard, we conducted two experiments to evaluate the recognition performance of summation invariants and other invariant functions. The first experiment addresses the problem of shape matching under uniform deformation. In the second experiment, we propose a more flexible and more practical shape-matching scheme where non-uniform deformation and occlusion are taken into account.

In chapter 4, we will discuss 3D face recognition by using summation invariants. The face recognition problem is currently an active research topic, and researchers in this field have proposed many approaches. First, we will present a literature survey with an emphasis on technical terminologies, popular techniques, and previous works on 3D face recognition. Second, we will introduce some facial image databases that are publicly available and widely used in the research community. Third, we will describe the proposed face recognition algorithm in detail and discuss the experimental results. Finally, we will make concluding remarks on this topic.

In chapter 5, we will present a multi-region face recognition algorithm and discuss some fusion schemes: namely, the sum rule, linear fusion, and nonlinear fusion.

Chapter 6 presents the summary of my dissertation and remarks on future works.

Chapter 2

Cartan's Moving Frame Method and Its Application to Object Recognition

2.1 Background

Among all the methods used to obtain invariants, there is one that stands out because of its simplicity, elegance and generality. This method is known as the *method of moving frames*. In the early nineteenth century, Élie Cartan [3] formulated this powerful tool for studying the geometric properties of manifolds and their invariants under the actions of transformation group. In the 1970's, several researchers began to place Cartan's intuitive constructions on a theoretical foundation. An important conceptual step was to define a *moving frame* as a cortain kind of mapping from a manifold to a transformation group. More recently, Fels et al. [19, 20] formulate a new systematical approach for applying the *method of moving frames* to general transformation group. This section provides a quick introduction of the basic ideas.

2.1.1 Lie Groups

A Lie group is a group manifold such that the associated group operation is smooth, *i.e.* infinitely differentiable. Two examples are given to illustrate its application.

Example 2.1 The general linear group $GL(n, \mathbb{R})$ consisting of all real invertible $n \times n$ matrices with matrix multiplication as group operation.

Example 2.2 Let's consider the Euclidean transformation acting on $\mathbb{R}^2$ where objects on a plane are subject to translation and rotation operations. Such a transformation may be described using linear matrix operation in homogeneous coordinates as follows:

$$\begin{bmatrix} \overline{x} \\ \overline{y} \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & a \\ sin(\theta) & cos(\theta) & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

where $\theta, a, b \in \mathbb{R}$. Therefore, the transformation matrices form a group with matrix multiplication as the group operation.

2.1.1.1 Transformation Groups

In many cases, transformations acting on space form a group. More specifically, we have the following:

Definition 2.1 Let M be a manifold. A *transformation group* acting on M is a Lie group G which satisfies

$$e \circ x = x, \quad g \circ (h \circ x) = (g \circ h) \circ x, \quad \forall x \in M, g \in G \tag{2.2}$$

, where e is an identity element of the Lie group G.

Example 2.3 The $n \times n$ matrices with nonzero determinant form a group by matrix multiplication on column vector $x \in \mathbb{R}^n$. This group is usually denoted by $GL(n, \mathbb{R})$.

2.1.2 Orbits and Invariant Functions

Let G be a Lie group acting on a manifold M. The orbit of a point $x \in M$ is the set of all images of x under group action. To be precise, the orbit of x is given by

$$\mathcal{O}_x = \{g \circ x : g \in G\} \tag{2.3}$$

An invariant for the transformation group G is a function $\eta : M \to \mathbb{R}$ which satisfies $\eta(g \circ x) = \eta(x), \forall g \in G, x \in M$.

Example 2.4 In the case of rotation acting on $\mathbb{R}^2$, the orbits are circles centered at the origin for any point $p \neq (0, 0)$. An invariant function $\eta : \mathbb{R}^2 \to \mathbb{R}$ is

$$\eta(x, y) = \sqrt{x^2 + y^2} \tag{2.4}$$

Example 2.5 In the case of translation acting on $\mathbb{R}^2$, orbits are the whole $\mathbb{R}^2$ plane. In this case, no nonconstant invariant function $\eta : \mathbb{R}^2 \to \mathbb{R}$ exists.

As illustrate by these two example, the dimension of orbit must be less than the dimension of manifold for us to be able to find an invariant function.

2.1.3 Jet Space and Prolongations

Unfortunately, it is quite common that the dimension of an orbit is greater than or equal to the dimension of the manifold, for example, the Euclidean group acting on the plane. One way to fix this problem is to create a larger space so that we can find an invariant function there. In this section, we will introduce a suitable space for this purpose, the so called *jet space*. We then define how group actions are *prolonged* so that the coordinates of jet space are appropriately transformed.

Traditionally, a jet space is coordinated by independent variables, dependent variables and derivatives of dependent variables. Hence, we call it *derivative jet space*. Consider a smooth function $u = f(x)$ which involves p independent variables $x = (x_1, \ldots, x_p)$ and q dependent variables $u = (u_1, \ldots, u_q)$. This allows us to define the *derivative jet space* J^n, whose coordinates consist of p independent variables, q dependent variables and all the partial derivatives of order up to n. A point in the derivative jet space J^n is denoted by $(x, u^{(n)})$, where $u^{(n)}$ contains dependent variables and partial derivatives up to order n. The action of group G on J^n is called the n^{th} *prolongation* and is denoted by $pr^n G$. This prolonged group action is defined so that the derivatives of function $u = f(x)$ are mapped to corresponding derivatives of transformed function $\overline{u} = \overline{f}(\overline{x})$. Specifically, for any point $(x_0, u_0^{(n)}) \in J^n$, the prolonged group action is defined by

$$pr^{(n)} g \circ (x_0, u_0^{(n)}) = (\overline{x}_0, \overline{u}_0^{(n)})$$

In other words, the transformed derivatives are found by evaluating the derivatives of transformed function $\overline{f}(\overline{x})$ at point $\overline{x}_0$.

Example 2.6 For the Euclidean group acting on $\mathbb{R}^2$, the first prolongation $pr^{(1)}G$ will act on the space coordinated by (x, u, u_x). By equation 2.1 and the definition of prolongation, the transformed coordinates are given by

$$\begin{aligned} \overline{x} &= x \cdot \cos\theta - u \cdot \sin\theta + a \\ \overline{u} &= x \cdot \sin\theta + u \cdot \cos\theta + b \\ \overline{u}_{\overline{x}} &= \frac{d\overline{u}}{dx}\frac{dx}{d\overline{x}} = \frac{\sin\theta + u_x \cos\theta}{\cos\theta - u_x \sin\theta} \end{aligned}$$

In other words, the prolonged group action is

$$pr^{(1)} g \circ (x, u, u_x) = (x \cos\theta - u \sin\theta + a, x \sin\theta + u \cos\theta + b, \frac{\sin\theta + u_x \cos\theta}{\cos\theta - u_x \sin\theta})$$

2.1.4 Method of Moving Frames

The invariant functions of a group G acting on a jet space J^n is defined as follows.

Definition 2.2 An *invariant* for a group G is a function $\eta : J^n \to \mathbb{R}$ such that

$$\eta(pr^{(n)}g \circ (x, u^{(n)})) = \eta((x, u^{(n)})), \quad \forall pr^{(n)}g \in pr^{(n)}G \text{ and } (x, u^{(n)}) \in J^n$$

In this section, we will introduce a systematic way to produce invariants of a Lie group G acting on a manifold M. The idea dating back to the early nineteenth century, due to Élie Cartan [3], is the theory of moving frames. It can be summarized by the procedures below,

> First, group action is prolonged onto derivative jet space. We can solve for group parameters using *normalization equations*. Then, substituting group parameters into un-normalized jet space coordinates gives us differential invariants.

Example 2.7 Let us illustrate the theory above with a simple example: Euclidean group acting on $\mathbb{R}^2$, shown in equation 2.1. For simplicity, let us assume the curve on the plane can be expressed as a smooth function $y = f(x)$. The corresponding derivative jet space $J^{(2)}$ is given by,

$$J^{(2)} = (x, y, y_x, y_{xx})$$

where y_x, y_{xx} denote the first and second derivatives respectively. The prolonged group action is

$$pr^{(2)}g \circ (x, y, y_x, y_{xx}) = (\overline{x}, \overline{y}, \overline{y}_{\overline{x}}, \overline{y}_{\overline{x}\overline{x}})$$

where

$$\overline{y}_{\overline{x}} = \frac{d\overline{y}}{d\overline{x}} = \frac{d\overline{y}}{dx}\frac{dx}{d\overline{x}} = \frac{\sin\theta + y_x\cos\theta}{\cos\theta - y_x\sin\theta}$$

By setting $\overline{x} = 0, \overline{y} = 0$ and $\overline{y}_{\overline{x}} = 0$, we can solve for $\{\theta, a, b\}$. This gives

$$\theta = -\arctan y_x, \quad a = \frac{-x - yy_x}{\sqrt{1+y_x^2}}, \quad b = \frac{xy_x - y}{\sqrt{1+y_x^2}}$$

Then, after substituting $\{\theta, a, b\}$ into $\overline{y}_{\overline{xx}}$ we have

$$\eta = \frac{y_{xx}}{\sqrt{1+y_x^2}} \tag{2.5}$$

Equation (2.5) is the fundamental differential invariant for the Euclidean group acting on $\mathbb{R}^2$. The equations in example 2.7, $\{\overline{x} = 0, \overline{y} = 0$ and $\overline{y}_{\overline{x}} = 0\}$,are called *normalization equations*. Actually, we have the freedom to choose any fixed point in derivative jet space. The guideline here is to choose a fixed point such that the normalization equations can be easily solved. In this example, $(0, 0, 0)$ is a natural choice.

The result in Equation (2.5) can be verified by observing its relationship with curvature and arc length. For a plane curve written in the form $y = f(x)$, the equation of curvature κ is given by

$$\kappa = \frac{y_{xx}}{(1+y_x^2)^{3/2}} = \frac{\eta}{1+y_x^2} = \frac{\eta}{s^2} \tag{2.6}$$

, where s is arc length. It's obvious that Euclidean transformations don't change curvature and arc length on a curve. Their product, $\eta = \kappa \cdot s^2$, is thus an Euclidean invariant.

From example 2.7, we can also make an intuitive explanation about how the *method of moving frames* works. Consider the blue curve shown in Figure 2.1, we draw tangent and normal vector at a chosen point on that curve. Solving *normalization equations* gives us a transformation matrix which brings that chosen point to the origin and makes the tangent vector at that point to be horizontal, as in the red curve in Figure 2.1. That transformation matrix, which is called the *moving frame,* also makes the normal vector at the chosen point vertical. So, no matter how the blue curve is rotated and translated, the normal vector transformed by the moving frame stays unchanged.

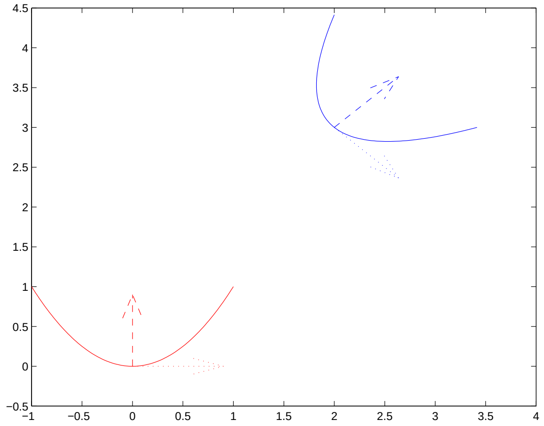


Figure 2.1 Method of moving frame: original curve (blue) and the curve transformed by moving frame (red).

The *method of moving frames* provides us a powerful and algorithmic tool to find invariant functions. But in practical applications, one is forced to compute differential invariants by a discrete numerical approximation and thus they are sensitive to noise.

2.2 Related Works

The fundamental problem about differential invariants is that high-order derivatives have to be computed, which amplify the effect of noise. For example, the simplest differential invariant of the equi-affine group requires fourth order derivatives. There have been several approaches developed to reduce the effect of noise. In this section, we introduce some of the popular approaches known in the literature.

2.2.1 Approximation of Differential Invariants

In order to reduce sensitivity to noise, *semi-differential invariants* were introduced by Van Gool et al. [13, 21]. In such an approach, a higher order differential invariant is approximated by a joint invariant depending on lower order derivatives evaluated at several points on a curve. Thus, the computation of high order derivatives can be avoided.

In [22], a robust differentiation technique has been developed. A *canonical* coordinate system which is independent of a given world coordinate system is defined so that all quantities defined on it are invariants. This method involves smoothing a discretized curve to make the result reasonably reliable. Human judgments are required to determine appropriate coefficients for smoothing.

Calabi *et. al.* [4] constructed numerical approximations of differential invariants by using successive points on a curve. They discussed the cases of planar curves under Euclidean and affine transformations. Their numerical approximation of Euclidean curvature depends on Euclidean distance between successive points and that of affine curvature depends on triangular areas of successive points. In my opinion, this is similar to avoiding computation of high order derivatives.

2.2.2 Invariant Parameterization

In [18], Sato and Cipolla introduce a new framework of integral invariants under transformation group action based on *invariant parameterization*. They work on an important case in many computer vision problems, affine transformation acting on $\mathbb{R}^2$. An affine quasi-invariant parameterization is derived. The idea behind it is to approximate affine arc-length by using low order derivatives.

Consider a curve C, to be parameterized by w. Let I denote a line integral of a function F along C on the interval $[w_1, w_2]$

$$I = \int_{w_1}^{w_2} F\, dw$$

They replace w with the *affine quasi-invariant arc-length* τ and choose $F(\tau) = \mu(\tau)^{\frac{-1}{2}}$, μ denoting affine curvature. The *affine quasi-invariant arc-length* can be described by Euclidean arc-length v and Euclidean curvature k as follows,

$$\tau = k^{\frac{2}{5}} dv$$

Since τ is computed from second order derivatives, their method reduces noise-induced fluctuations by avoiding the computation of high order derivatives.

2.2.3 Integral Invariant Signatures

It's obvious that distance and area are unaffected by Euclidean transformation. Based on this fact, Manay *et al.* [23] propose two integral invariants with respect to Euclidean transformation. Given a curve in $\mathbb{R}^2$, they define a disk centered at a point on that curve. On the disk, the *integral distance invariant* computes distance metrics and the *integral area invariant* computes area metrics. By moving the disk and computing these two invariants at each point, they can plot the *integral invariant signature*.

Unlike the method of moving frames, this method can not systematically produce invariants. Also, the generalization of their method to other transformation groups is not trivial.

2.2.4 Integral Invariant

Hann and Hickman [11] observe that the root of the problem lies in the derivation of invariants. The traditional approach prolongs group actions to derivatives and thus the resulting invariants will depend on derivatives. They define a new jet space, which is coordinatized by integrals. Thus, the resulting invariants will depend on integrals rather than derivatives and so will not be sensitive to noise. Compared with the previous two approaches, their method does not use any derivatives and can generate invariants systematically. They open a new window for the equivalence problem of shapes under transformation group actions. Since their work is particularly important, we will discuss it in some detail.

2.2.4.1 Extending a Lie Group Action on $\mathbb{R}^2$ to Potentials

Consider a Lie group G acting on $\mathbb{R}^2$ defined by

$$g \circ (x, u) = (\overline{x}, \overline{u}), g \in G$$

The conventional way of deriving invariants is to prolong the group action to *derivative jet space* J. For example, $J = (x, u, u_x, u_x x)$. For integral invariants, the goal is to extend group action to potentials rather than derivatives. The definition of potentials and potential jet space are shown below.

Definition 2.3 The **potential** $V^{i,j}$ of order k is given by

$$V^{i,j} = \int x^i u^j dx \tag{2.7}$$

, where $j \neq 0$ and $i + j = k$.

Definition 2.4 The **potential jet space** J_p^n is coordinatized by $(x_1, u_1, x_0, u_0, V^{(n)})$.

(x_0, u_0) and (x_1, u_1) denote the initial and end points of a curve, respectively. $V^{(n)}$ consists of potentials up to n^{th} order.

For example,

$$\begin{aligned} J_p^1 &= (x_1, u_1, x_0, u_0, V^{0,1}) \\ J_p^2 &= (x_1, u_1, x_0, u_0, V^{0,1}, V^{1,1}, V^{0,2}) \end{aligned}$$

2.2.4.2 Affine Invariant

Consider the affine group acting on $\mathbb{R}^2$,

$$g \circ (x, u) = (ax + bu + c, dx + eu + f), \quad \det \begin{pmatrix} a & b \\ d & e \end{pmatrix} \neq 0$$

To find invariants, we first solve for the *moving frame*, $\{a, b, c, d, e, f\}$, by setting

$$(\overline{x}_1, \overline{u}_1, \overline{x}_0, \overline{u}_0, \overline{V}^{0,1}, \overline{V}^{1,1}) = (0, 0, 1, 1, 0, 0)$$

Then, an integral invariant $\eta_{integral}$ can be found by substituting $\{a, b, c, d, e, f\}$ into $\overline{V}^{0,2}$.

$$\begin{aligned}\eta_{integral} &= \big\{ -3(x_1-x_0)V^{0,2}+6(u_1-u_0)V^{1,1}+4(V^{0,1})^2 \\ &\quad -2V^{0,1}(2x_1u_1-2x_0u_0+x_0u_1-x_1u_0)+u_0u_1(x_1-x_0)^2\big\} \\ &\quad /\big\{((x_1-x_0)(u_1+u_0)-2V^{0,1})^2\big\} \qquad (2.8)\end{aligned}$$

Given a curve $u = f(x)$ through (x_0, u_0), $\eta_{integral}$ can be computed at each point (x, u). Note that if two curves C and $\overline{C}$ are related by an affine transformation, a point $(x_0, u_0) \in C$ must be identified with the corresponding point $(\overline{x}_0, \overline{u}_0) \in \overline{C}$ to evaluate $\eta_{integral}$.

Integral invariants are different from differential invariants in the sense that differential invariants are defined locally. They are also different from the moment invariants [7], which are defined globally. Integral invariants are defined semi-locally because the range of integration can be specified by varying (x_0, u_0) and (x_1, u_1). In other words, integral invariants can extract local characteristics like differential invariants do. Meanwhile, unlike moment invariants, they don't suffer from occlusion problems. This property gives us more flexibility in dealing with practical problems.

In practical object recognition problems, we are usually given sets of discrete points. Each set of discrete points represents a shape and we want to classify these shapes. From Hann and Hickman's point of view, they assume there are continuous shapes behind the discrete data, which allows them to compute the integral invariants. Their idea has one obvious drawback. It will highly depend on the sampling resolution. The higher the sampling rate is, the more accurate the integral will be.

2.3 Summation Invariants

In this dissertation, we develop a novel scheme to find invariants, which are based on the summation operation of discrete data, called *summation invariants*. They are independent of sampling rate and maintain all the nice properties of integral invariants, such as the systematic production of invariants and high noise immunity. The transformation groups used to illustrate these novel techniques are the Euclidean and affine groups, with emphasis on the cases of curves and surfaces in two and three-dimensional space.

In the following section, we give definitions of *potential* and *potential jet space*. Then, summation invariants for curves in $\mathbb{R}^2$ and for surfaces in $\mathbb{R}^3$ are explicitly derived. A variety of fundamental geometric transformations are discussed in detail.

2.3.1 Summation Invariants for Curves in $\mathbb{R}^2$

Suppose we have the boundary of a planar object, which is parameterized as $\{(x_n, y_n) : n = 1, 2, \ldots, N\}$. Note that the parameterization is not necessarily equally spaced. Consider a transformation group G acting on $\mathbb{R}^2$ defined by

$$g \circ (x_n, y_n) = (\overline{x}_n, \overline{y}_n), g \in G$$

We define *potential* and *potential jet space* as follows.

Definition 2.5 The **potential** $P_{i,j}$ of order k is given by

$$P_{i,j} = \sum_{n=1}^{N} x_n^i \cdot y_n^j, \quad \text{where } i + j = k$$

Definition 2.6 The **potential jet space** J^k is the Euclidean space with coordinates

$$(x_1, y_1, x_N, y_N, P_{(k)})$$

where $P_{(k)}$ consists of potentials up to k^{th} order.

For example,

$$J^1 = (x_1, y_1, x_N, y_N, \sum_{n=1}^{N} x_n, \sum_{n=1}^{N} y_n)$$

In the following two sections, we will address two important transformation groups in computer vision, namely Euclidean and affine transformations, and derive their summation invariants.

2.3.1.1 Euclidean Summation Invariant of Curves

Given the sampled points on a curve $\{(x_n, y_n) : n = 1, 2, \ldots, N\}$, we use $(\bar{x}_n, \bar{y}_n)$ to denote that it is transformed by a Euclidean transformation.

$$\begin{bmatrix} \bar{x}_n \\ \bar{y}_n \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.9}$$

where $a, b, \theta \in \mathbb{R}$.

One can construct a family of invariant functions $\eta_{i,j}$ by applying a *moving frame* to potentials $P_{i,j}$. For example, we can find a moving frame by solving the following equations,

$$(\bar{x}_1, \bar{y}_1, \bar{y}_N) = (0, 0, 0) \tag{2.10}$$

The equations above are called *normalization equations*. Note that we are free to specify normalization equations as long as they can be solved, *i.e.* a moving frame can be found. Here, we specify the normalization equations so that the correspodning moving frame can be easily solved. Let $\bar{P}_{i,j}$ be the potentials transformed by a moving frame,

$$\bar{P}_{i,j} = \sum_{n=1}^{N} \bar{x}_n^i \bar{y}_n^j \tag{2.11}$$

where $\bar{x}$ and $\bar{y}$ denote the x and y coordinates transformed by a moving frame. The $\bar{P}_{i,j}$ are invariant functions under Euclidean transformation acting on $\mathbb{R}^2$, *i.e.* $\eta_{i,j} = \bar{P}_{i,j}$. The first and second order invariant functions, $i + j = 1$ or 2, have been explicitly derived as shown below:

$$\begin{aligned}\eta_{1,0} &= P_{1,0}(x_N - x_1) + P_{0,1}(y_N - y_1)\\ &\quad + Nx_1(x_1 - x_N) + Ny_1(y_1 - y_N)\end{aligned}\tag{2.12}$$

$$\begin{aligned}\eta_{0,1} &= P_{1,0}(y_N - y_1) + P_{0,1}(x_1 - x_N)\\ &\quad + N(x_N y_1 - x_1 y_N)\end{aligned}\tag{2.13}$$

$$\begin{aligned}\eta_{2,0} &= -2P_{1,0}(x_1 - x_N)(x_1^2 - x_1x_N + y_1^2 - y_1y_N)\\ &\quad -2P_{0,1}(y_1 - y_N)(y_1^2 - y_1y_N + x_1^2 - x_1x_N)\\ &\quad +P_{2,0}(x_1 - x_N)^2 + P_{0,2}(y_1 - y_N)^2\\ &\quad +2P_{1,1}(x_1 - x_N)(y_1 - y_N)\\ &\quad +N(x_1(x_1 - x_N) + y_1(y_1 - y_N))^2\end{aligned}\tag{2.14}$$

$$\begin{aligned}\eta_{1,1} &= P_{1,1}((x_1 - x_N)^2 - (y_1 - y_N)^2)\\ &\quad +P_{1,0}(y_1^3 + 2x_1x_Ny_N - 2y_Nx_1^2\\ &\quad +x_1^2y_1 - 2y_1^2y_N + y_1y_N^2 - x_N^2y_1)\\ &\quad -P_{0,1}(x_1^3 + 2y_1y_Nx_N - 2x_Ny_1^2\\ &\quad +y_1^2x_1 - 2x_1^2x_N + x_1x_N^2 - y_N^2x_1)\\ &\quad +(P_{0,2} - P_{2,0})(x_1 - x_N)(y_1 - y_N)\\ &\quad +N(x_Ny_1 - x_1y_N)(x_1(x_N - x_1) + y_1(y_N - y_1))\end{aligned}\tag{2.15}$$

$$\begin{aligned}\eta_{0,2} &= 2(x_Ny_1 - x_1y_N)(P_{1,0}(y_N - y_1) - P_{0,1}(x_N - x_1))\\ &\quad +P_{2,0}(y_1 - y_N)^2 + P_{0,2}(x_1 - x_N)^2\\ &\quad -2P_{1,1}(x_1 - x_N)(y_1 - y_N) + N(x_1y_N - x_Ny_1)^2\end{aligned}\tag{2.16}$$

2.3.1.2 Affine Summation Invariant of Curves

Affine transformation can be used to approximate the perspective projection of 3D objects on the image plane. Therefore, invariants of affine transformation have significant importance in computer vision. In this section, we explicitly derive the summation invariants of affine transformations. Consider the affine transformation group G acting on $\mathbb{R}^2$ given by,

$$\begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \tag{2.17}$$

where $\det \begin{bmatrix} a & b \\ d & e \end{bmatrix} \neq 0$

Again, we prolong group action to potentials. The affine-transformed potentials are shown below:

$$\overline{P}_{1,0} = \sum_{n=1}^{N}(ax_n + by_n + c) = a \cdot P_{1,0} + b \cdot P_{0,1} + c \cdot N \tag{2.18}$$

$$\overline{P}_{0,1} = \sum_{n=1}^{N}(dx_n + ey_n + f) = d \cdot P_{1,0} + e \cdot P_{0,1} + f \cdot N \tag{2.19}$$

Next, *moving frame* $\{a, b, c, d, e, f\}$ can be found by solving the *normalization equations*. In this case, we specify the *normalization equations* as follows:

$$(\overline{x}_1, \overline{y}_1, \overline{x}_N, \overline{y}_N, \overline{P}_{1,0}, \overline{P}_{0,1}) = (0, 0, 1, 1, 0, 0) \tag{2.20}$$

By applying the *moving frame* $\{a, b, c, d, e, f\}$ to higher order potentials, it will yield summation invariants under affine group actions. The second order potentials are shown below:

$$P_{2,0} = \sum_{n=1}^{N} x_n^2 \tag{2.21}$$

$$P_{1,1} = \sum_{n=1}^{N} x_n \cdot y_n \tag{2.22}$$

$$P_{0,2} = \sum_{n=1}^{N} y_n^2 \tag{2.23}$$

They are transformed by affine group action as follows,

$$\begin{aligned}
\overline{P}_{2,0} &= a^2P_{2,0} + 2abP_{1,1} + 2acP_{1,0} + 2bcP_{0,1} + b^2P_{0,2} + c^2N && (2.24)\\
\overline{P}_{1,1} &= adP_{2,0} + (ae+bd)P_{1,1} + (af+cd)P_{1,0} + (bf+ce)P_{0,1} + beP_{0,2} + cfN && (2.25)\\
\overline{P}_{0,2} &= d^2P_{2,0} + 2deP_{1,1} + 2dfP_{1,0} + 2efP_{0,1} + e^2P_{0,2} + f^2N && (2.26)
\end{aligned}$$

Then, we apply *moving frame* $\{a, b, c, d, e, f\}$ solved from Equation (2.20) to $\overline{P}_{2,0}$. The resulting affine summation invariant $\zeta_{2,0}$ is given by,

$$\begin{aligned}
\zeta_{2,0} &= \big\{P_{2,0}(Ny_1 - P_{0,1})^2 + P_{0,2}(Nx_1 - P_{1,0})^2\\
&\quad -2P_{1,1}(Nx_1 - P_{1,0})(Ny_1 - P_{0,1}) - N(y_1P_{1,0} - x_1P_{0,1})^2\big\}\\
&\quad /\big\{(N(x_Ny_1 - x_1y_N) + (y_N - y_1)P_{1,0} - (x_N - x_1)P_{0,1})^2\big\} && (2.27)
\end{aligned}$$

Note that we can find infinitely many affine invariant functions by substituting solved values of $\{a, b, c, d, e, f\}$ into higher order transformed potentials.

Furthermore, one can show that

$$\begin{aligned}
g \circ \text{denominator}(\zeta_{2,0}) &= (ae - bd)^2\text{denominator}(\zeta_{2,0}) && (2.28)\\
g \circ \text{numerator}(\zeta_{2,0}) &= (ae - bd)^2\ \text{numerator}(\zeta_{2,0}) && (2.29)
\end{aligned}$$

In other words, both denominator and numerator of $\zeta_{2,0}$ are relative invariants under affine transformation. From an application point of view, relative invariants provide enough information for object recognition and we don't need to worry about the situation where the denominator vanishes.

2.3.2 Summation Invariants for Surfaces in $\mathbb{R}^3$

Recent developments in 3D sensor technologies have accelerated the collection of 3D models. Nonetheless, computational problems arise when comparing these data. In this section, we will develop the summation invariants for transformation groups acting on $\mathbb{R}^3$.

Consider a surface $S \subseteq \mathbb{R}^3$. We can treat it as a mapping from a connected region $\mathcal{U} \subseteq \mathbb{R}^2$ to $\mathbb{R}^3$

$$S : (u, v) \longmapsto (x, y, z), \text{where } (u, v) \in \mathcal{U} \tag{2.30}$$

In many engineering fields, we only have a set of sampled points on a surface S. Here, we assume it is represented as a discrete function with two independent variables.

$$S = \begin{bmatrix} x[m,n] \\ y[m,n] \\ z[m,n] \end{bmatrix}, \text{ where } m \in \{1, \ldots, M\} \text{ and } n \in \{1, \ldots, N\}$$

Before prolongation of group action, a new definition for *potentials* is required.

Definition 2.7 The **potential** $Q_{i,j,k}$ of order ℓ is given by

$$Q_{i,j,k} = \sum_{m=1}^{M} \sum_{n=1}^{N} x^i[m,n] \cdot y^j[m,n] \cdot z^k[m,n] \tag{2.31}$$

, where i, j, k are nonnegative integers, $i + j + k = \ell$ and $\ell \neq 0$.

Then, the summation invariants of a surface patch under transformation group actions can be derived by Cartan's method of moving frames.

2.3.2.1 Euclidean Summation Invariants of Surfaces

In this section, we consider a Euclidean transformation acting on $\mathbb{R}^3$. Given a parameterized surface $(x[m,n], y[m,n], z[m,n])$ with $m = 1, \ldots, M$ and $n = 1, \ldots, N$, the summation invariants $\kappa_{i,j,k}$ of surfaces under Euclidean transformation can be expressed in a compact form

$$\kappa_{i,j,k} = \bar{Q}_{i,j,k} = \sum_{m=1}^{M} \sum_{n=1}^{N} \bar{x}^i \cdot \bar{y}^j \cdot \bar{z}^k \tag{2.32}$$

where $\bar{x}, \bar{y}$ and $\bar{z}$ denote x, y and z coordinates transformed by a moving frame. The remaining task is simply to find a moving frame. In the previous section, we find a moving frame by solving normalization equations. Here, solving normalization equations is not trivial since the Euclidean transformation group acting on $\mathbb{R}^3$ has more degrees of freedom than the one acting on $\mathbb{R}^2$. Instead of solving normalization equations, a moving frame can be expressed in terms of rotation matrices $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$ and a translation vector $\mathbf{T}$.

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \mathbf{R}_3\mathbf{R}_2\mathbf{R}_1(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \mathbf{T}) \tag{2.33}$$

We choose a moving frame such that

$$(\bar{x}[1,1], \bar{y}[1,1], \bar{z}[1,1], \bar{y}[M,1], \bar{z}[M,1], \bar{z}[1,N]) = (0,0,0,0,0,0) \tag{2.34}$$

Hence, we have

$$\mathbf{T} = \begin{bmatrix} x[1,1] \\ y[1,1] \\ z[1,1] \end{bmatrix} \tag{2.35}$$

so that $(\bar{x}[1,1], \bar{y}[1,1], \bar{z}[1,1]) = (0,0,0)$. Then, we need to fix the other coordinates in Equation (2.34). Let

$$\mathbf{A} = \begin{bmatrix} x[M,1] \\ y[M,1] \\ z[M,1] \end{bmatrix} \tag{2.36}$$

and express $\mathbf{A}$ in spherical coordinates (R_a, θ_a, ϕ_a), where θ and ϕ are angular displacements measured from the positive z-axis and positive x-axis, respectively, and R is the distance from the origin to the point. By setting

$$\mathbf{R}_1 = \begin{bmatrix} \cos\phi_a & \sin\phi_a & 0 \\ -\sin\phi_a & \cos\phi_a & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.37}$$

$$\mathbf{R}_2 = \begin{bmatrix} \sin\theta_a & 0 & \cos\theta_a \\ 0 & 1 & 0 \\ -\cos\theta_a & 0 & \sin\theta_a \end{bmatrix} \tag{2.38}$$

, the rotation matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ will rotate $\mathbf{A}$ so that it is lying on the x-axis, *i.e.*

$$(\bar{y}[M,1], \bar{z}[M,1]) = (0,0) \tag{2.39}$$

Now, there is one degree of freedom left to be fixed. Let

$$\mathbf{B} = \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = \mathbf{R}_2\mathbf{R}_1(\begin{bmatrix} x[1,N] \\ y[1,N] \\ z[1,N] \end{bmatrix} - \mathbf{T}) \tag{2.40}$$

and

$$\mathbf{C} = \begin{bmatrix} B_y \\ B_z \\ B_x \end{bmatrix} \tag{2.41}$$

Again, the vector $\mathbf{C}$ can be expressed in spherical coordinates (R_c, θ_c, ϕ_c). By setting

$$\mathbf{R}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_c & \sin\phi_c \\ 0 & -\sin\phi_c & \cos\phi_c \end{bmatrix} \tag{2.42}$$

, the rotation matrices $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$ and translation $\mathbf{T}$ form a moving frame. Now, the summation invariants $\kappa_{i,j,k}$ can be generated by simply applying the moving frame to the potentials $Q_{i,j,k}$.

2.3.2.2 Affine Summation Invariants of Surfaces

The affine group acting on $\mathbb{R}^3$ can be described as below,

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} j \\ k \\ \ell \end{bmatrix} = \begin{bmatrix} \overline{x} \\ \overline{y} \\ \overline{z} \end{bmatrix} \tag{2.43}$$

Once again, we prolong group action to potentials. After applying group action, $Q_{1,0,0}$ is given by

$$\begin{aligned} \overline{Q}_{1,0,0} &= \sum_{m=1}^{M}\sum_{n=1}^{N} \overline{x}[m,n] \\ &= \sum_{m=1}^{M}\sum_{n=1}^{N} (ax + by + cz + j) \\ &= aQ_{1,0,0} + bQ_{0,1,0} + cQ_{0,0,1} + jMN \end{aligned} \tag{2.44}$$

Similarly,

$$\begin{aligned}
\overline{Q}_{0,1,0} &= \sum_{m=1}^{M}\sum_{n=1}^{N}\overline{y}[m,n] \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N}(dx+ey+fz+k) \\
&= dQ_{1,0,0}+eQ_{0,1,0}+fQ_{0,0,1}+kMN \qquad (2.45)
\end{aligned}$$

$$\begin{aligned}
\overline{Q}_{0,0,1} &= \sum_{m=1}^{M}\sum_{n=1}^{N}\overline{z}[m,n] \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N}(gx+hy+iz+\ell) \\
&= gQ_{1,0,0}+hQ_{0,1,0}+iQ_{0,0,1}+\ell MN \qquad (2.46)
\end{aligned}$$

Next, *moving frame* $\{a,b,c,d,e,f,g,h,i,j,k,\ell\}$ cab be found by solving *normalization equations*. In this case, we specify *normalization equations* as follows:

$$\begin{aligned}
&(\overline{x}[1,1],\overline{y}[1,1],\overline{z}[1,1],\overline{x}[M,1],\overline{y}[M,1],\overline{z}[M,1], \\
&\overline{x}[1,N],\overline{y}[1,N],\overline{z}[1,N],\overline{Q}_{1,0,0},\overline{Q}_{0,1,0},\overline{Q}_{0,0,1}) \\
&=(0,0,0,1,0,0,0,1,0,0,0,0) \qquad (2.47)
\end{aligned}$$

By applying the solved parameters $\{a,b,c,d,e,f,g,h,i,j,k,\ell\}$ to affine-transformed potentials $\overline{Q}_{i,j,k}$, one can obtain invariant functions $\Psi_{i,j,k}$ under the affine group acting on $\mathbb{R}^3$. To demonstrate the feasibility of the method of moving frames, the affine invariant $\Psi_{2,0,0}$ will be explicitly derived here. Other affine invariants $\Psi_{i,j,k}$ can also be explicitly derived in exactly the same way. The affine-transformed potential $\overline{Q}_{2,0,0}$ is given by,

$$\begin{aligned}\overline{Q}_{2,0,0} &= \sum_{m=1}^{M}\sum_{n=1}^{N}\overline{x}^2[m,n] \\ &= a^2Q_{2,0,0}+b^2Q_{0,2,0}+c^2Q_{0,0,2}+2j(aQ_{1,0,0}+bQ_{0,1,0}+cQ_{0,0,1}) \\ &\quad +2abQ_{1,1,0}+2bcQ_{0,1,1}+2acQ_{1,0,1}+j^2MN\end{aligned} \tag{2.48}$$

By substituting the parameters $\{a, b, c, d, e, f, g, h, i, j, k, \ell\}$, obtained by solving Equation (2.47), into $\overline{Q}_{2,0,0}$, the resulting affine invariant $\Psi_{2,0,0}$ of surface is given by:

$$\begin{aligned}\Psi_{2,0,0} =&\{Q_{0,0,2}(MN(x_{01}y_{00}-x_{00}y_{01})+Q_{100}(y_{01}-y_{00})+Q_{010}(x_{00}-x_{01}))^2 \\ &-2Q_{011}(MN(x_{01}y_{00}-x_{00}y_{01})+Q_{100}(y_{01}-y_{00})+Q_{010}(x_{00}-x_{01})) \\ &(MN(x_{01}z_{00}-x_{00}z_{01})+Q_{100}(z_{01}-z_{00})+Q_{001}(x_{00}-x_{01})) \\ &+Q_{020}(MN(x_{01}z_{00}-x_{00}z_{01})+Q_{100}(z_{01}-z_{00})+Q_{001}(x_{00}-x_{01}))^2 \\ &-2Q_{110}(MN(x_{01}z_{00}-x_{00}z_{01})+Q_{100}(z_{01}-z_{00})+Q_{001}(x_{00}-x_{01})) \\ &(MN(y_{01}z_{00}-y_{00}z_{01})+Q_{010}(z_{01}-z_{00})+Q_{001}(y_{00}-y_{01})) \\ &+Q_{200}(MN(y_{01}z_{00}-y_{00}z_{01})+Q_{010}(z_{01}-z_{00})+Q_{001}(y_{00}-y_{01}))^2 \\ &-2Q_{101}(MN(x_{01}y_{00}-x_{00}y_{01})+Q_{100}(y_{01}-y_{00})+Q_{010}(x_{00}-x_{01})) \\ &(MN(y_{00}z_{01}-y_{01}z_{00})+Q_{010}(z_{00}-z_{01})+Q_{001}(y_{01}-y_{00})) \\ &-MN(Q_{100}(y_{01}z_{00}-y_{00}z_{01})+Q_{010}(x_{00}z_{01}-x_{01}z_{00})+Q_{001}(x_{01}y_{00}-x_{00}y_{01}))^2\} \\ &/(MN(x_{00}(y_{10}z_{01}-y_{01}z_{10})+x_{10}(y_{01}z_{00}-y_{00}z_{01})+x_{01}(y_{00}z_{10}-y_{10}z_{00})) \\ &+Q_{100}(y_{00}(z_{01}-z_{10})+y_{10}(z_{00}-z_{01})+y_{01}(z_{10}-z_{00})) \\ &+Q_{010}(x_{00}(z_{10}-z_{01})+x_{10}(z_{01}-z_{00})+x_{01}(z_{00}-z_{10})) \\ &+Q_{001}(x_{00}(y_{01}-y_{10})+x_{10}(y_{00}-y_{01})+x_{01}(y_{10}-y_{00})))^2\end{aligned} \tag{2.49}$$

, where $x_{00} = x[1,1], y_{00} = y[1,1], z_{00} = z[1,1], x_{10} = x[M,1], y_{10} = y[M,1], z_{10} = z[M,1], x_{01} = x[1,N], y_{01} = y[1,N], z_{01} = z[1,N]$.

Note that the summation invariant is quite different from the global invariants, such as moment invariants [7]. The moment invariants require the whole shape. On the contrary, summation invariants can be evaluated within any region on a surface.

Under affine group actions, the numerator and denominator of $\Psi_{2,0,0}$ are changed by

$$\begin{aligned} \text{numerator}(\overline{\Psi}_{2,0,0}) &= (ceg - bfg - cdh - afh + bdi - aei)^2 \text{numerator}(\Psi_{2,0,0}) \\ \text{denominator}(\overline{\Psi}_{2,0,0}) &= (ceg - bfg - cdh - afh + bdi - aei)^2 \text{denominator}(\Psi_{2,0,0}) \end{aligned}$$

This shows that the numerator and denominator of affine invariant $\Psi_{2,0,0}$ are relative affine invariants. Recall that the same behavior is observed on the affine summation invariant $\zeta_{2,0}$. This is an important observation since relative affine invariants provide enough information for classification. Performance improvement can also be achieved by combining multiple invariants at various levels. We will address the issue of integrating classifiers in a separate chapter.

2.3.3 A Numerical Example

In this example, two surfaces related by a rigid transformation are generated and their corresponding summation invariants are calculated. The goal is to give an intuitive understanding of the behavior of the summation invariant and show that perfect invariance is preserved under noise-free condition.

Consider a surface S given by

$$S = \begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix} = \begin{bmatrix} sin(u)cos(v) \\ sin(u)sin(v) \\ cos(u) \end{bmatrix}$$

The data points on surface S are generated by uniformly sampling along u from 0.1π to 0.5π and along v from 0 to 0.5π. Another surface $\tilde{S}$ is produced by applying a rigid transformation on the surface S. The relation between the surface S and the transformed surface $\tilde{S}$ is given by:

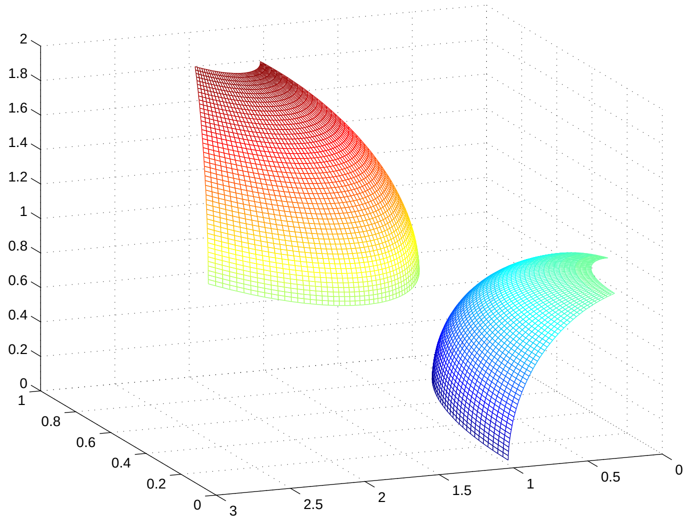


Figure 2.2 Two surfaces related by Euclidean transformation

$$\begin{bmatrix} cos(\frac{3\pi}{4}) & -sin(\frac{3\pi}{4}) & 0 \\ sin(\frac{3\pi}{4}) & cos(\frac{3\pi}{4}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix}$$

These two surfaces are shown in Figure 2.2. Note that rigid transformations consist of rotations and translations, *i.e.* no scaling or shearing. In other words, the transformation matrix of a rigid transformation always has determinant equal to 1. Under such a circumstance, one could use either denominator or numerator of $\Psi_{2,0,0}$ and both of them will yield perfect invariance. Here, both denominator and numerator of $\Psi_{2,0,0}$ are computed from two surfaces, shown below.

$$\begin{aligned} \text{numerator}(\Psi_{2,0,0}(x, y, z)) &= 1.3141 \times 10^7 \\ \text{denominator}(\Psi_{2,0,0}(x, y, z)) &= 5.8731 \times 10^4 \\ \text{numerator}(\Psi_{2,0,0}(\tilde{x}, \tilde{y}, \tilde{z})) &= 1.3141 \times 10^7 \\ \text{denominator}(\Psi_{2,0,0}(\tilde{x}, \tilde{y}, \tilde{z})) &= 5.8731 \times 10^4 \end{aligned}$$

Evidently, both denominator and numerator of $\Psi_{2,0,0}$ remain unchanged when the surface S is under an arbitrary rigid transformation.

2.4 Robustness in the Presence of Noise

Without the presence of noise, invariant functions obtained by different approaches all exhibit perfect invariance properties. However, it is always the case in practical situations that the invariance properties will not be exactly maintained and one can only compute invariant features with limited precision. Thus, it is of great importance to analyze the robustness to noise for different kinds of invariant functions. In this section, we will compare the summation invariants and the integral invariants in terms of their numerical robustness. In particular, theoretical analysis relating second-order statistics are derived to evaluate the impact of noise.

Suppose one wants to evaluate a function f of several variables $\mathbf{X}_1, \ldots, \mathbf{X}_n$. The variables $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are only available in quantized form, *i.e.* we only have access to $q(\mathbf{X}_1), \ldots, q(\mathbf{X}_n)$ where the quantization function is defined by

$$q(x) = k\Delta_q \qquad \text{,if } (k-0.5)\Delta_q \leq x < (k+0.5)\Delta_q \tag{2.50}$$

where Δ_q is the quantization interval and k is an integer. Consequently, the error in the computation of f is

$$\Delta f = f(\mathbf{X}_1, \ldots, \mathbf{X}_n) - f(q(\mathbf{X}_1), \ldots, q(\mathbf{X}_n)) \tag{2.51}$$

Thus, we are only concerned with the error due to quantization of the value of variables and are not concerned with error due to computation or evaluation of functions. In principle, we can compute the moments of Δf if the joint probability distribution of $\mathbf{X}_1, \ldots, \mathbf{X}_n$ is known. Specifically, the n^{th} moment of Δf is given by

$$M_n = \int \cdots \int (\Delta f)^n h_{\mathbf{X}_1,\ldots,\mathbf{X}_n}(x_1, \ldots, x_n) dx_1 \cdots dx_n \tag{2.52}$$

where $h_{\mathbf{X}_1,\ldots,\mathbf{X}_n}$ is the joint probability density function of $\mathbf{X}_1, \ldots, \mathbf{X}_n$. However, in practice we will see that f can be any linear or nonlinear function of its variables and an analytical evaluation of quantization error can be difficult. Some assumptions is usually made to simplify the error analysis. We will present the error analysis of summation invariants and integral invariants in the following two sections. For each, its robustness is characterized in terms of the variance of error in the computed value, as described by Equation (2.51).

2.4.1 Error Analysis on Summation Invariants

In this section, we will give the analytic derivation of the expected squared error for two simple cases, namely $\eta_{0,1}$ and $\eta_{1,0}$. This serves to give an idea of the mathematics involved. For other summation invariants, their error analysis can be done in the same way. The focus is placed on $\eta_{0,1}$ and $\eta_{1,0}$ because they are composed of second order monomials. In the following section, the integral invariants will be composed of second order monomials, too. This leads to a fair comparison between them in terms of their numerical stability.

First, it is assumed without loss of generality that quantization errors are independent identically distributed random variables. In order to simplify the analysis, it is further assumed that the mean of x coordinates and the mean of y coordinates are not zero, *i.e.*

$$m_x = \frac{1}{N} \sum_{n=1}^{N} x_n \neq 0 \tag{2.53}$$

$$m_y = \frac{1}{N} \sum_{n=1}^{N} y_n \neq 0 \tag{2.54}$$

Then,

$$\begin{aligned} \frac{\sum_{n=1}^{N} q(x_n)}{\sum_{n=1}^{N} x_n} &= \frac{\sum_{n=1}^{N} x_n + \epsilon_{x_n}}{\sum_{n=1}^{N} x_n} \\ &= 1 + \frac{\sum_{n=1}^{N} \epsilon_{x_n}}{\sum_{n=1}^{N} x_n} \\ &= 1 + \frac{\sum_{n=1}^{N} \epsilon_{x_n}}{N m_x} \end{aligned} \tag{2.55}$$

where ϵ_{x_n} is the quantization error associated with x_n. One can easily show that,

$$\lim_{N \to \infty} \frac{\sum_{n=1}^{N} q(x_n)}{\sum_{n=1}^{N} x_n} = 1 \tag{2.56}$$

Based on this, we can assume N is large enough so that the potential $P_{0,1}$ and $P_{1,0}$ are relatively insensitive to the quantization error, *i.e.*

$$\sum_{n=1}^{N} q(x_n) \approx \sum_{n=1}^{N} x_n \tag{2.57}$$

$$\sum_{n=1}^{N} q(y_n) \approx \sum_{n=1}^{N} y_n \tag{2.58}$$

Therefore, the error in the estimate $\hat{\eta}_{0,1}$ is given by

$$\begin{aligned} \Delta\eta_{0,1} &= \hat{\eta}_{0,1} - \eta_{0,1} \\ &\approx -(y_N - y_1) \sum_{n=0}^{N-1} \epsilon_{x_n} - (\epsilon_{y_N} - \epsilon_{y_1}) \sum_{n=0}^{N-1} x_n \\ &\quad +(x_N - x_1) \sum_{n=0}^{N-1} \epsilon_{y_n} + (\epsilon_{x_N} - \epsilon_{x_1}) \sum_{n=0}^{N-1} y_n \\ &\quad -N\{x_N \epsilon_{y_1} + y_1 \epsilon_{x_N} - x_1 \epsilon_{y_N} - y_N \epsilon_{x_1}\} \end{aligned} \tag{2.59}$$

One can compute the expected value and the variance of $\Delta\eta_{0,1}$.

$$\begin{aligned} E[\Delta\eta_{0,1}] &= 0 & (2.60) \\ var[\Delta\eta_{0,1}] &\approx \sigma^2\{(\sum_{n=1}^{N} y_n - Ny_N)^2 + (\sum_{n=1}^{N} y_n - Ny_1)^2 \\ &\quad +(\sum_{n=1}^{N} x_n - Nx_N)^2 + (\sum_{n=1}^{N} x_n - Nx_1)^2 \\ &\quad +N(y_N - y_1)^2 + N(x_N - x_1)^2\} & (2.61) \end{aligned}$$

Here, the observations are the following. First, the quantization error does not introduce bias in the estimate $\hat{\eta}_{0,1}$. Second, the variance of the estimate $\hat{\eta}_{0,1}$ is proportional to N^2.

Similarly, the error in the estimate of $\eta 1, 0$ is given by

$$\begin{aligned} \Delta\eta_{1,0} &= \hat{\eta}_{1,0} - \eta_{1,0} \\ &\approx -(\epsilon_{x_N} - \epsilon_{x_1})P_{1,0} - (\epsilon_{y_N} - \epsilon_{y_1})P_{0,1} \\ &\quad -N[-x_1\epsilon_{x_N} + \epsilon_{x_1}(2x_1 - x_N) - y_1\epsilon_{y_N} + \epsilon_{y_1}(2y_1 - y_N)] & (2.62) \end{aligned}$$

where the products of the quantization errors are regarded as negligible. The mean and variance of $\Delta\eta_{1,0}$ are given by

$$\begin{aligned} E[\Delta\eta_{1,0}] &= 0 & (2.63) \\ var[\Delta\eta_{1,0}] &= \sigma^2\{(-P_{1,0} + N(2x_1 - x_N))^2 + (P_{1,0} - Nx_1)^2 + \\ &\quad (-P_{0,1} + N(2y_1 - y_N))^2 + (P_{0,1} - Ny_1)^2\} & (2.64) \end{aligned}$$

Here, the observations are the same as the previous case, *i.e.* quantization error does not introduce bias and the variance is proportional to N^2.

2.4.2 Error Analysis on Integral Invariants

Hann and Hickman [11] introduced the *integral invariants* and explicitly derive the affine integral invariant Equation (2.8). Unfortunately, it is not easy to give an analytical error analysis on Equation (2.8). Instead we consider a simpler case, Euclidean/rigid transformation group, and derive the corresponding integral invariants. Then, the mean and variance of the estimate of the Euclidean/rigid integral invariants can be derived analytically.

Let $\{(x, y) : y = f(x), x \in [x_a, x_b]\}$ be a curve parameterized by x. In order to find integral invariants, one need to first define a *jet space* J. For Euclidean group acting on $\mathbb{R}^2$, the jet space is given by

$$J = \{(x_a, y_a, y_b) : y_a = f(x_a), y_b = f(x_b)\} \tag{2.65}$$

Then, one can specify normalization equations and find the corresponding moving frame. To make the algebraic manipulation as easy as possible, we choose the normalization equations to be

$$(\bar{x}_a, \bar{y}_a, \bar{y}_b) = (0, 0, 0) \tag{2.66}$$

where $(\bar{x}_a, \bar{y}_a, \bar{y}_b)$ denotes the results of applying an Euclidean transformation to (x_a, y_a, y_b). After solving the normalization equations, integral invariants can then be obtained by applying the moving frame to potentials. Recall that the potentials defined by Hann and Hickman [11] are shown in Definition 2.3.

Thus, $V^{0,1}$ is an order 1 monomial potential and $V^{1,1}, V^{0,2}$ are order 2 monomial potentials. The integral invariant shown below is obtained by applying the moving frame to $V^{0,1}$.

$$\mu_{0,1} = V^{0,1} - \frac{(x_b - x_a)(y_b + y_a)}{2} \tag{2.67}$$

The most straightforward methods for numerical integrations are the left and right Riemann sums. Here, we use trapezoidal sums as a numerical approximation. The reason for this choice will be explained later. The trapezoid approximation of $V^{0,1}$ associated with the partition $x_a = x_0 < x_1 < \ldots < x_{N-1} = x_b$ is given by

$$\hat{V}^{0,1} = \frac{1}{2}\sum_{n=0}^{N-2}(y_{n+1} + y_n)(x_{n+1} - x_n) \tag{2.68}$$

The estimate $\hat{\mu}_{0,1}$ is calculated by replacing $V^{0,1}$ in $\mu_{0,1}$ with $\hat{V}^{0,1}$. Note that the estimate $\hat{\mu}_{0,1}$ itself is also an invariant function, *i.e.*

$$\hat{\mu}_{0,1}(x, y) = \hat{\mu}_{0,1}(\bar{x}, \bar{y}) \tag{2.69}$$

where $\bar{x}, \bar{y}$ denotes a curve under arbitrary Euclidean/rigid transformations. Put

$$\nu_{0,1} = -2 \cdot \hat{\mu}_{0,1} \tag{2.70}$$

and it can be easily shown that

$$\nu_{0,1} = \sum_{n=0}^{N-1}(x_n y_{n+1} - x_{n+1} y_n) \tag{2.71}$$

where the indices of x, y are given modulo N. Following Eq 2.71 and Eq 2.51, the error in the integral invariants $\nu_{0,1}$ due to the quantization in x and y coordinates is given by

$$\begin{aligned} \Delta\nu_{0,1} &= \sum_{n=0}^{N-1} x_n y_{n+1} - x_{n+1} y_n - \sum_{n=0}^{N-1} q(x_n)q(y_{n+1}) - q(x_{n+1})q(y_n) \\ &= \sum_{n=0}^{N-1} x_n y_{n+1} - x_{n+1} y_n - \\ &\quad \sum_{n=0}^{N-1} (x_n + \epsilon_{x_n})(y_{n+1} + \epsilon_{y_{n+1}}) - (x_{n+1} + \epsilon_{x_{n+1}})(y_n + \epsilon_{y_n}) \end{aligned} \tag{2.72}$$

It is assumed without loss of generality that the $\epsilon_{x_n}, \epsilon_{y_n}$ are independent and identically distributed, with zero mean and variance σ^2. Since the products of $\epsilon_{x_n}, \epsilon_{y_n}$ are relatively small, the quantization error $\Delta\nu_{0,1}$ is approximated by its first-order errors, *i.e.*

$$\Delta\nu_{0,1} \approx \sum_{n=0}^{N-1} x_n\epsilon_{y_{n+k}} + y_{n+k}\epsilon_{x_n} - x_{n+k}\epsilon_{y_n} - y_n\epsilon_{x_{n+k}} \tag{2.73}$$

Thus, the expected value and the variance of $\Delta\nu_{0,1}$ are given by

$$E[\Delta\nu_{0,1}] = 0 \tag{2.74}$$

$$var[\Delta\nu_{0,1}] = \sigma^2\Big(\sum_{n=0}^{N-1}(x_n - x_{n+2k})^2 + (y_n - y_{n+2k})^2\Big) \tag{2.75}$$

According to Equation (2.74), the quantization error does not introduce bias in the estimate $\hat{\nu}_{0,1}$. Also, the variance of the estimate $\hat{\nu}_{0,1}$ is proportional to N and the quantization parameter σ.

2.4.3 Experiment Results

A numerical simulation is conducted to evaluate the accuracy of the derived first and second-order statistics. The simulation consists of the following steps:

1. Generate a semi circle with radius 100 and perform sampling uniformly along its arc length. The number of sampling points is 200.

2. Pick up the first N points of the semi circle and compute its invariants. The initial value of N is 30.

3. The curve with N points is subject to arbitrary Euclidean transformations and then quantization. The parameters of Euclidean transformation are randomly generated. During the quantization, x and y coordinates are rounded to the nearest integer. Then, we compute invariants from the quantized curve. This step is repeated 1000 times.

4. Increase N by 10 and repeat steps 2 and 3.

In practical applications, shapes are always given in quantized format. Hence, there is no way for a user to control the quantization granularity. Reflecting on this fact, the quantization granularity is a fixed value in our simulation. We should also point out that the value of an invariant feature depends on the shape of an object. Here, the analysis is limited to the cases where quantization errors are relatively small, *i.e.* $\eta \gg \Delta\eta$. For the shapes not satisfying this condition, the computed invariant values are not reliable and hence have little practical value.

Figure 2.3 shows the variance of $\Delta\nu_{0,1}$. It is observed that the results of simulation are very close to that of the theoretical analysis. Also, the variance is proportional to N as predicted by Equation (2.75). The variances of $\Delta\eta_{0,1}$ are shown in Figure 2.4. Again, the results of the simulation are very close to the theoretical predictions.

Figure 2.5 shows the variance of relative errors which is a more descriptive quantity in assessing the precision of an estimate. It is immediately clear that relative error is inversely proportional to N. So, one can increase the accuracy of summation or integral invariants by increasing N. According to the error analysis in the previous section, the variances of $\Delta\nu_{0,1}$ and $\Delta\eta_{0,1}$ are proportional to N and N^2 respectively. Hence, the variance of the relative error $\frac{\Delta\nu_{0,1}}{\nu_{0,1}}$ is expected to drop faster than that of $\frac{\Delta\eta_{0,1}}{\eta_{0,1}}$. This phenomenon is observed in the results of the simulation. It indicates that $\nu_{0,1}$ can reduce error more efficiently than $\eta_{0,1}$ does.

However, we should also point out that choosing a large N has an undesirable consequence. In particular, the local characteristics of a shape will be averaged by using a large N. The rule of thumb is to keep N as small as possible while maintaining an acceptable accuracy. According Figure 2.5, $\eta_{0,1}$ and $\nu_{0,1}$ achieve similar accuracy in the range where N is relatively small.

Thus, one may conclude that 1% mean squared relative error can be achieved for curve segments with 30 points by using either $\eta_{0,1}$ or $\nu_{0,1}$.

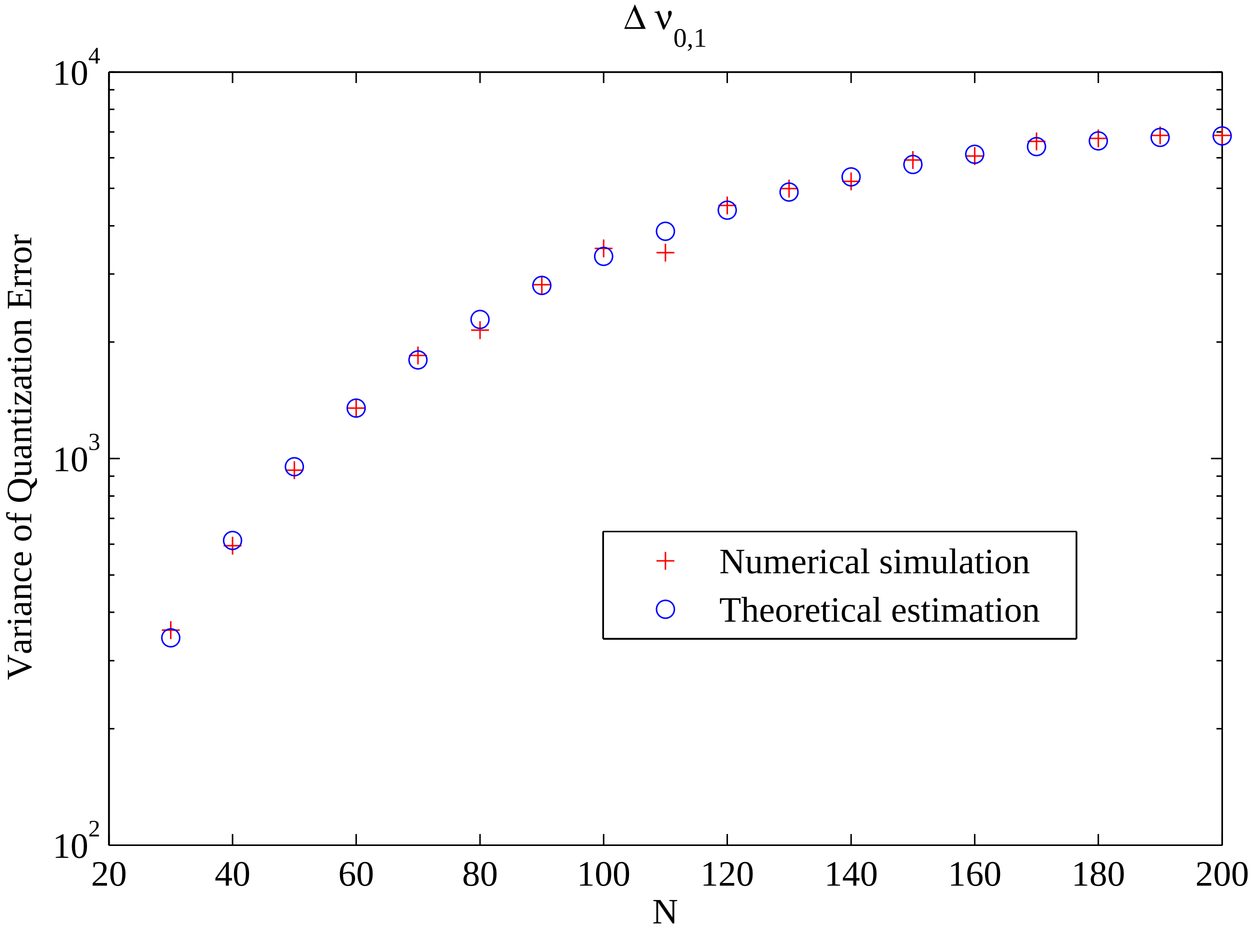


Figure 2.3 The variance of quantization error $\Delta\nu_{0,1}$

2.5 Comparison with Curvature

In vision problems, the need to compute curvature from a discretized curve arises quite frequently. However, it is also well known in the computer vision community that achieving an accurate curvature estimation is a rather challenging task. In this section, we will discuss the source of errors when computing curvatures. This serves mainly to give a feeling for why this is a difficult problem.

In differential geometry [24], the definition of curvature is given by

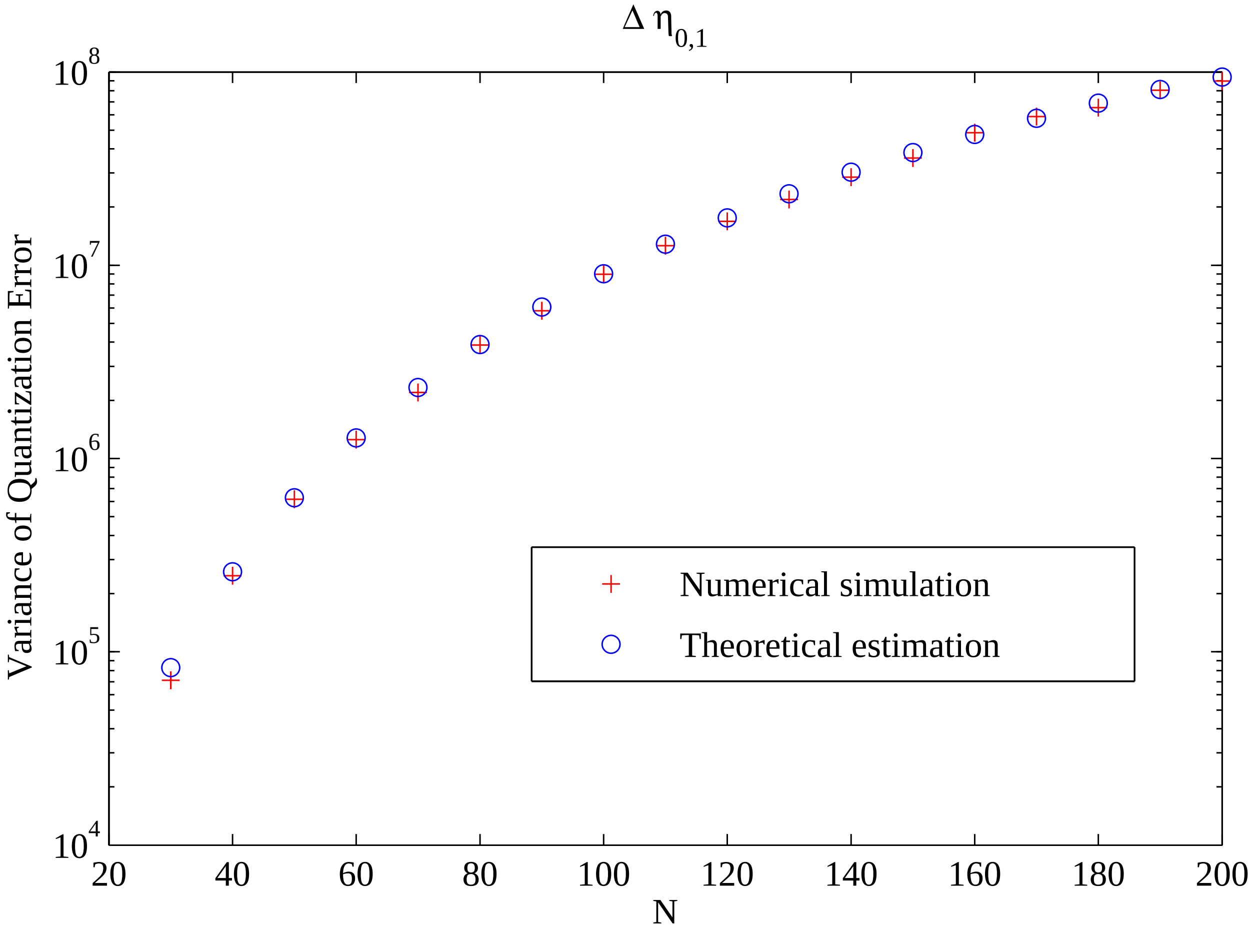


Figure 2.4 The variance of quantization error $\Delta\eta_{0,1}$

.

Definition 2.8 Let $\alpha : [a, b] \to \mathbb{R}^2$ be a curve parameterized by arc length $s \in [a, b]$. The curvature κ of α at s is defined as

$$\frac{d\vec{t}}{ds} = \kappa\vec{n} \tag{2.76}$$

where $\vec{t}$ and $\vec{n}$ are the tangent and normal vectors respectively.

According to Equation (2.76), curvature is proportional to the second order derivative. Hence, the problem of estimating the curvature has much in common with that of estimating the derivatives

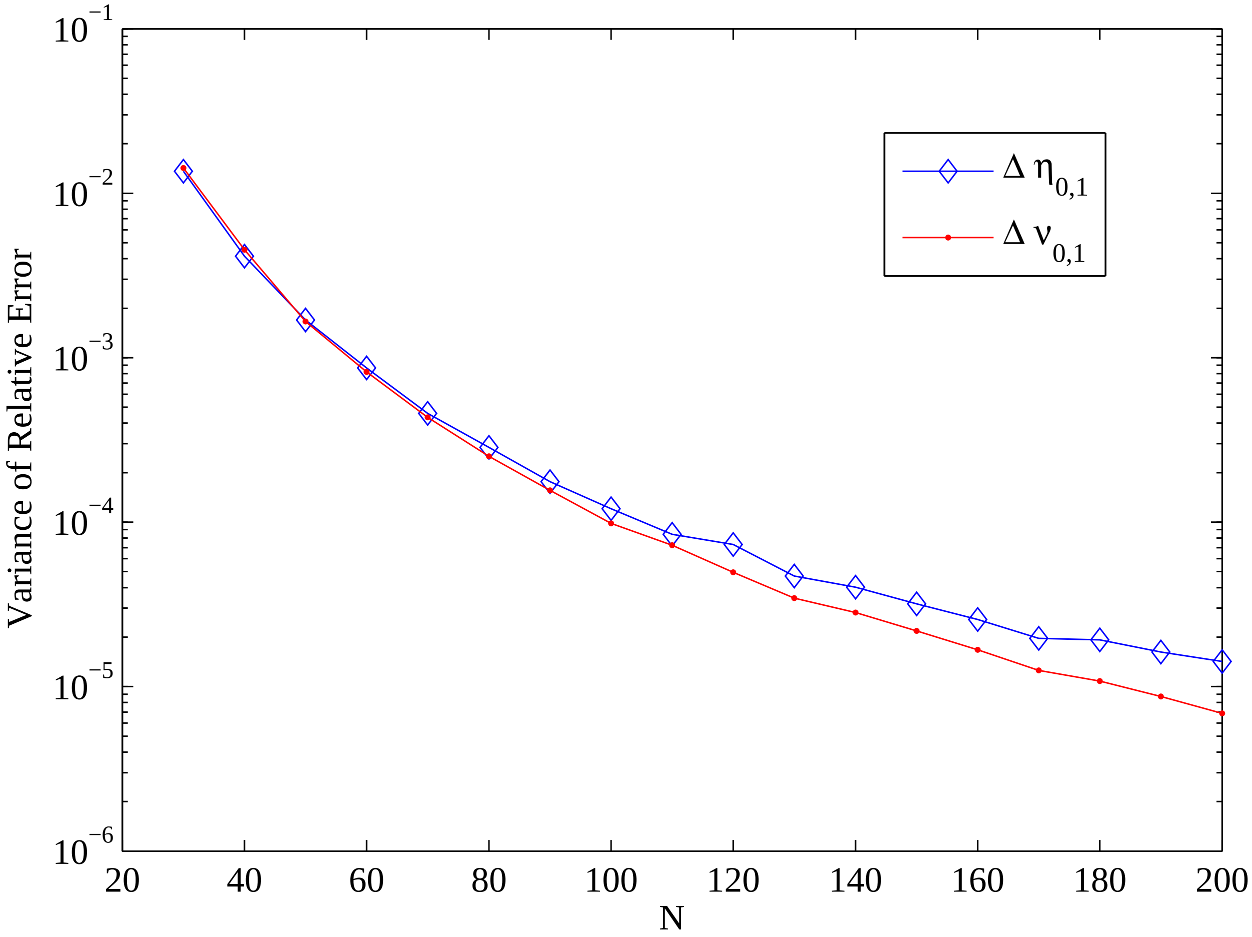


Figure 2.5 The variance of relative error $\frac{\Delta\eta_{0,1}}{\eta_{0,1}}$ and $\frac{\Delta\nu_{0,1}}{\nu_{0,1}}$

.

of a function. For simplicity, we will illustrate the issue of estimating derivatives by analyzing a very simple case.

Consider a real valued function $f : \mathbb{R} \rightarrow \mathbb{R}$, its first derivative is the limit of Equation (2.77) as dx approaches zero.

$$\hat{f'} = \frac{f(x + dx) - f(x)}{dx} \tag{2.77}$$

According to the definition, the intuition is that to achieve the highest accuracy of numerical computation, one should compute Equation (2.77) with as small a value of dx as possible. However, this yields erroneous results under practical situations.

To investigate the source of error, consider the following expression for estimating the first derivative.

$$\begin{aligned} \hat{f}'(x) &= \frac{f(x+dx)+\epsilon_1-f(x)-\epsilon_2}{dx} \\ &= \frac{f(x+dx)-f(x)}{dx}+\frac{\epsilon_1-\epsilon_2}{dx} \end{aligned}$$

where ϵ_1 and ϵ_2 are the errors in the values of $f(x+dx)$ and $f(x)$ respectively. If the errors are bounded by a maximum, ϵ_{max}, the worst case error in the estimate $\hat{f}'(x)$ is given by

$$\frac{2\epsilon_{max}}{dx} \tag{2.78}$$

To further simplify the analysis, ϵ_{max} is assumed to be independent of dx. Under this assumption, it is clear that the worst case error will become unacceptably large as dx gets smaller and smaller. On the other hand, it looks like one can reduce error bound by increasing dx. However, another source of error should be taken into account. According to Equation (2.77), $\hat{f}'$ could be considered as the average derivative in the interval $[x, x+dx]$, which is generally different from the value of the derivative at x. By increasing dx, one not only decreases the error bound but also decrease the precision of the location of the estimate. This phenomenon can be described by the Heisenberg uncertainty principle in quantum physics: the greater the precision of the position of the estimate, the smaller the precision of the estimate itself. For curvature as well as higher order derivatives, this is the fundamental issue one has to deal with.

In [25], the author made the following conclusion:

> the accurate calculation of the curvature for low-curvature regions is in fact impossible for common images sizes, while reasonable results may under favorable conditions be obtained for high-curvature regions.

Similarly, Kovalevsky [26] made similar comment about the estimation of the curvature.

> We can demonstrate that it is rather difficult to reach practically acceptable estimates of the curvatures in binary images.

On the contrary, summation invariants get around the difficulty posed by the uncertainty principle since they are defined on an interval $[x, x + dx]$ rather than a point. Unlike Equation (2.77) which is generally different from the derivative at x, the summation invariants provide perfect invariance under noise free condition. In the presence of noise, one can improve the accuracy of the summation invariants by increasing dx without losing the accuracy of the position.

2.6 Summary

The summation invariants are a substantial extension of the seminal work by Hann and Hickman [11] in the following ways:

- We propose the new definitions of potentials for planar curves and surface patches. The potentials defined in [11] have one major drawback: they tend to generate complicated normalization equations and thus finding the corresponding moving frames is difficult. For transformation groups acting on three dimensional space, solving the normalization equations become mathematically intractable. By revising the definition of potentials, the normalization equations are greatly simplified and thus finding invariants for transformation groups acting on three dimensional space becomes possible.
- Hann and Hickman [11] only discuss the case where the affine group acts on $\mathbb{R}^2$ and derive the corresponding affine integral invariants. However, generalization of their approach to

3D cases is nontrivial. In this dissertation, our approach is successfully applied in many important cases, including the Euclidean group acting on $\mathbb{R}^2$, affine group acting on $\mathbb{R}^2$, Euclidean group acting on $\mathbb{R}^3$ and affine group acting on $\mathbb{R}^3$. In each case, the corresponding summation invariants are explicitly derived.

Even though the essential philosophy behind summation invariants is similar to that of integral invariants, these two approaches exhibit quite different behavior in practical applications. For instance, given two set of points related by a particular geometric transformation, their corresponding summation invariants will be exactly the same under noise free condition. Note that we make no assumption about how data points are sampled, so it can be applied to a wide range of sampling conditions. In contrast, perfect invariance is not preserved by integral invariants due to numerical integration. Inevitably, integral invariants have to live with errors introduced by numerical approximation. This difference seems little but it turns out to make big difference in our experiments.

The issue of numerical robustness is investigated by conducting theoretical analysis and numerical experiments. For short curve segments (less than 60 points), both summation and integral invariants achieve an acceptable precision of 1% mean squared relative error. Compared to the numerical computation of curvature, summation invariants demonstrate a significantly better stability. For curvature, it has been shown that a precision higher than 50% can rarely be achieved under typical conditions [26].

In addition, the summation invariants share many nice properties of the differential and integral invariants, which are summarized below:

1. Infinitely many invariants can be derived by substituting moving frame into higher order potentials.

2. Extension to other transformation groups is straightforward.

3. The precision of a summation invariant can be improved by increasing the range of summation. In addition, one can perform multi-scale analysis by varying the range of the summation.

4. Summation invariants are local descriptors which allow shape matching under occlusion.

Chapter 3

2D Shape Matching Under Visual Deformation

The recognition of shapes in images is one of the fundamental issues in many computer vision applications. It is challenging due to the large space of shape deformations. The robustness of a shape representation to such variations can have significant impact on recognition performance. In this chapter, we will give a brief literature review on the problem of 2D shape recognition. Also, we will discuss the strengths and weakness of existing invariants used for object recognition and will give a comparison with summation invariants. Then, we will present a novel approach for 2D shape matching using geometric properties, summation invariants in particular.

Two experimental results are presented here. In the first experiment, 2D affine summation invariants are utilized for recognizing fish contours, *i.e. closed curves*. In the second experiment, we will consider a more generic situation, the matching of curve segments.

3.1 Introduction

In many 2D shape recognition applications, it is necessary to have a robust alignment so that corresponding features can be compared with each other. Many existing approaches for curve alignment utilize application-specific constraints, such as object recognition based on silhouettes [27–29], hand written character recognition [30, 31], etc. In contrast, we tackle the problem by introducing a novel family of geometric features which can be generically combined with application-specific constraints.

The curve alignment approaches can be roughly divided into two categories: methods based on rigid transformation [27, 32] and those based on non-rigid transformation [33–36]. On one hand, the approaches based on rigid transformations rely on finding optimal transformation parameters. The applicability of these approaches is relatively limited because of the assumption that shapes are only subject to rigid transformation. On the other hand, approaches based on non-rigid transformations solve the correspondence problem by minimizing a cost functional consisting of deformation metrics [33–36].

Since the method of moving frames is a systematic approach, the summation invariants can be easily tailored for both rigid and non-rigid cases. In the chapter, the goal is to demonstrate how summation invariants could be utilized for finding the optimal alignment between two curves.

3.2 Representation of Curves using Summation Invariants

Given a set of sample points on a curve, one can compute summation globally, *i.e.* the resulting invariant feature is a scalar number. However, it will be very difficult to perform accurate classification in such a low-dimensional feature space since it does not adequately capture local variations of the underlying objects. To address this problem, we propose a novel *semi-local summation invariant* feature that computes the summation invariant over a local interval of a curve rather than the entire curve. An important observation is that the lower and upper limits in the definition of the potentials are not functions of group operations. As such, these limits can be changed from all the points in the objects to a local subset of points. Let us use the affine invariants as an example to illustrate this method. For a curve $(x_n, y_n), n \in \{1, 2, \ldots, N\}$, the m^{th} semi-local summation affine invariant is given by

$$\lambda_{affine}[m] = (M(x_{m_M} y_{m_1} - x_{m_1} y_{m_M}) + P_{1,0}(y_{m_M} - y_{m_1}) - P_{0,1}(x_{m_M} - x_{m_1}))^2 \tag{3.1}$$

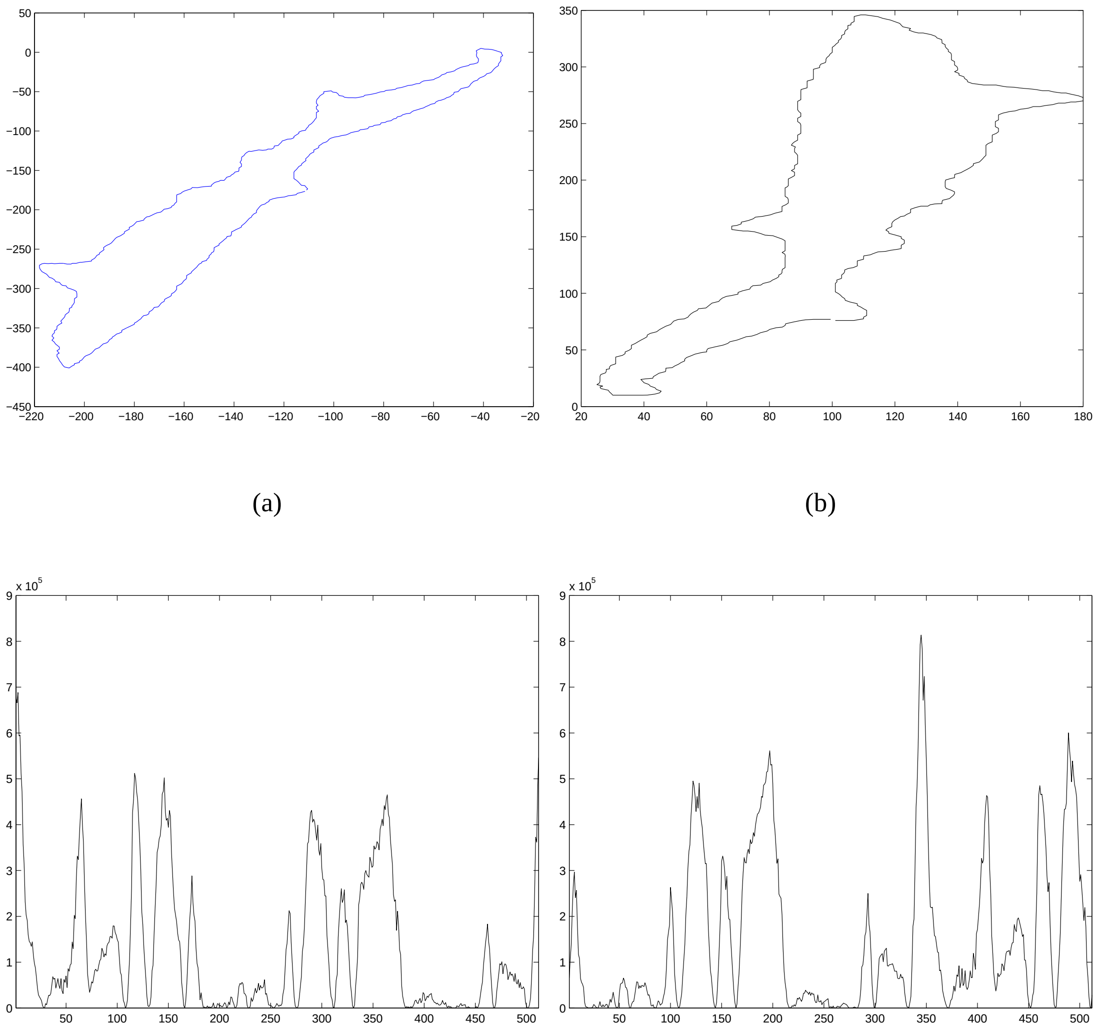
where

$$\begin{aligned} P_{1,0} &= \sum_{n=m}^{m+M-1} x[\text{mod}(n, N)] \\ P_{0,1} &= \sum_{n=m}^{m+M-1} y[\text{mod}(n, N)] \\ m_1 &= m \\ m_M &= \text{mod}(m + M - 1, N) \end{aligned}$$

Note that the major differences between these definitions and those of the global summation invariants are the different summation limits. We only use the denominator of η_{affine} because it is a *relative affine invariant*. Actually, one can choose either the denominator or numerator of η_{affine}, since they will work in the same manner.

Given N sampled points on a curve, one can compute semi-local summation invariants with non-overlapping or overlapping intervals. Obviously, using overlapping intervals will yield high-dimensional features. For simplicity, we will use overlapping intervals. Note that techniques for dimensionality reduction can be subsequently applied. In the following experiment, the semi-local summation invariant is computed from a local interval surrounding each sample point so that the resulting representation of a curve is a vector with dimension N. The length of the the local interval, M, which can be adjusted according to the specific application, is empirically determined.

Figure 3.1(a) and 3.1(b) show fish contours related by affine transformations. Their representations using semi-local summation invariants $\lambda_{affine}[m]$ are depicted in Figures 3.1(c) and 3.1(d) respectively. For a closed contour, there is neither a beginning point nor an ending point. For the shape representation using semi-local summation invariant, the location of a starting point only causes a cyclical translation as one can see in Figures 3.1(c) and 3.1(d).



(a) (b)

(c) (d)

Figure 3.1 (a) and (b) are two fish contours related by affine transformation. Their semi-local summation invariants $\lambda_{affine}[m]$ are shown in (c) and (d) respectively. Note (c) and (d) are related by translation because the initial point on the fish contour is randomly chosen.

3.3 Shape Matching Experiments Using Synthesized Data

In this section, several experiments are conducted to explore the behaviors of the shape representation using semi-local summation invariants. These experiments will accomplish three things.

First, they will illustrate how the proposed 2D shape representation could be applied to shape matching/recognition problems. Second, they will measure its performance compared to that of two alternative invariant features, the integral invariant feature [11], and the wavelet-based invariant features [10]. Third, the reliability of the proposed 2D shape representation in the presence of additive noise will be assessed.

3.3.1 Generating the Shape Dataset

A synthesized dataset of marine animals is created for the shape matching experiment. A subset of 100 distinct types of fish contours are selected from the SQUID database [37]. These selected fish contours are depicted in Figure 3.2 and are called fish prototypes. Note that some of them are rather similar. For example, 35 and 56, 37 and 45, as well as 40 and 51 all have similar shapes. Twenty deformations are generated for each prototype using affine transformations. The affine transform parameters $\{a, b, c, d, e, f\}$ are randomly generated for each deformation.

$$\begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \tag{3.2}$$

The affine transformed coordinates $\begin{bmatrix} \overline{x} & \overline{y} \end{bmatrix}^T$ will be quantized into integer values. If different points on the original contour are mapped onto the same coordinate after the affine transformation and quantization, the duplicate points are eliminated. Therefore, the total number of points in the affine-transformed curve may be different from that of the original curve. In Figure 3.3, the 20 variations of fish type 1 are depicted. We apply linear interpolation to resample each transformed contour so that all re-sampled contours have the same number of sample points. In this experiment, the number of sample points on each re-sampled contour is chosen to be 512. The relationship between affine-transformed contour $\overline{x}[n]$ and re-sampled contour $\overline{x}'[n]$ can be described as shown below:

$$\overline{x}'[n] = \begin{cases} \overline{x}[0] & \text{if } n = 0; \\ \overline{x}[\lfloor \frac{Nn}{512} \rfloor](\lceil \frac{Nn}{512} \rceil - \frac{Nn}{512}) + \overline{x}[\lceil \frac{Nn}{512} \rceil](\frac{Nn}{512} - \lfloor \frac{Nn}{512} \rfloor) & \text{if } n = 1, 2, \ldots, 511. \end{cases}$$

where N is the total number of sample points on an affine-transformed contour $\overline{x}[n]$. $\lfloor\ \rfloor$ and $\lceil\ \rceil$ are floor and ceiling operators, respectively.

Three different representations of a fish contour are computed by using semi-local affine summation invariant, semi-local affine integral invariant and affine wavelet invariant.

To calculate the semi-local summation invariant features, we choose $M = 51$ as the range of local summation. The resulting representation of a fish contour is a 512×1 feature vector. Each entry of this vector corresponds to $\lambda_{affine}[m]$, Equation (3.1), calculated from a 51-point shape segment.

It is important to know how the summation invariant performs compared with the integral invariant [11]. Like semi-local summation invariant, each fish contour is represented by a 512×1 feature vector. Each element of this feature vector is calculated by using the denominator of $\eta_{integral}$, Equation (2.8). The integration, $V^{0,1}$, is approximated by numerical integration over a shape segment with 51 points.

It's also important to know how it performs compared with the state-of-the-art methods. In [10], wavelet-based invariant functions were demonstrated to have superior discrimination power compared to some traditional methods, namely the moment invariant [8] and Fourier descriptor method [9]. We calculate $\eta_{3,4,5,6,7,8}[n]$, $n = 1, \ldots, 512$ using Daubechies' wavelet of length 4. According to the author's suggestion in [10], only the coarsest scale levels $(3, \ldots, 8)$ are used to calculate the affine invariant function. The finest scale levels have been dropped because they are sensitive to noise. As a result, each fish contour is represented by a 512×1 feature vector.

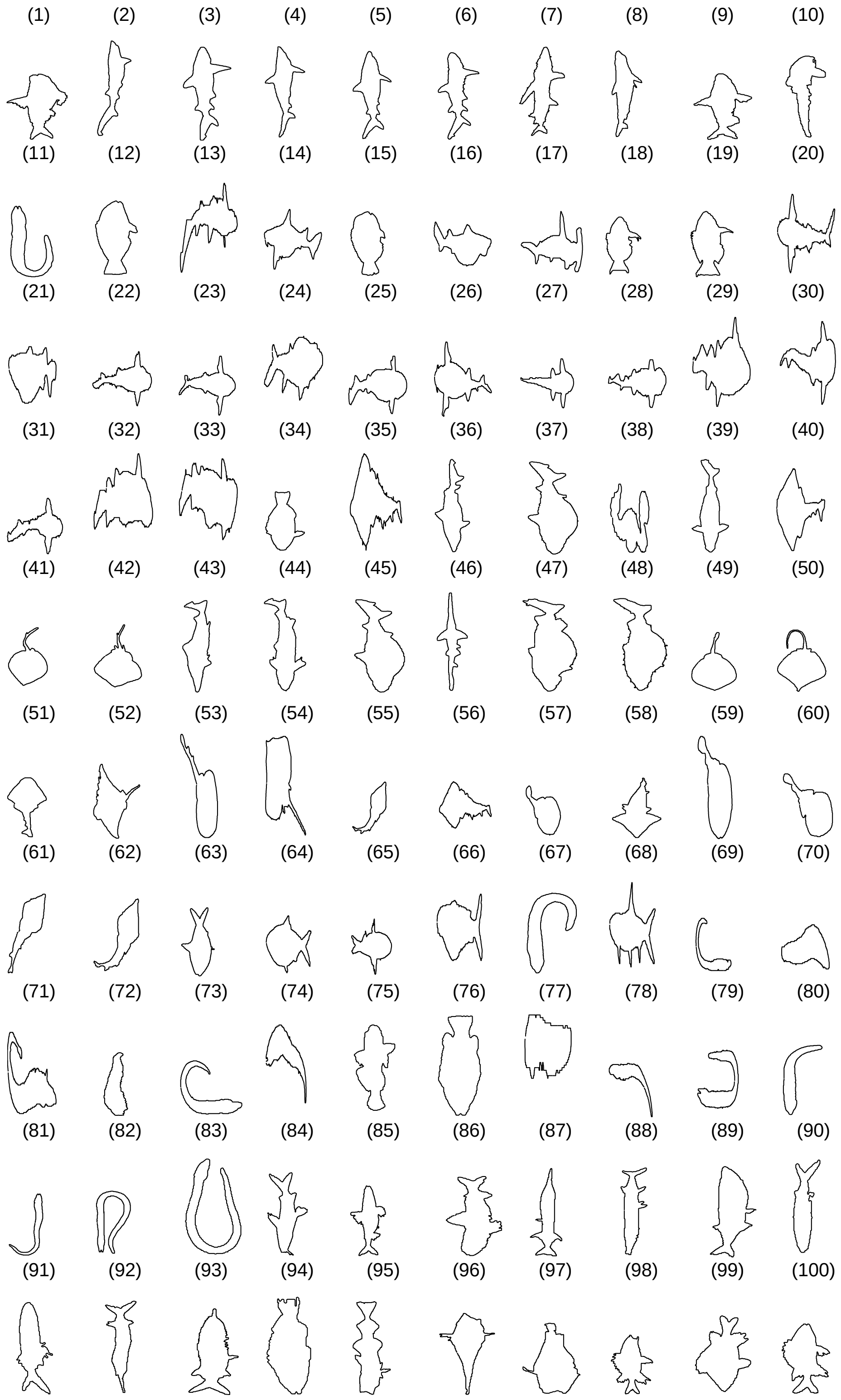


Figure 3.2 Database contains 100 prototype fish contours. Each prototype represents one distinct type of fish.

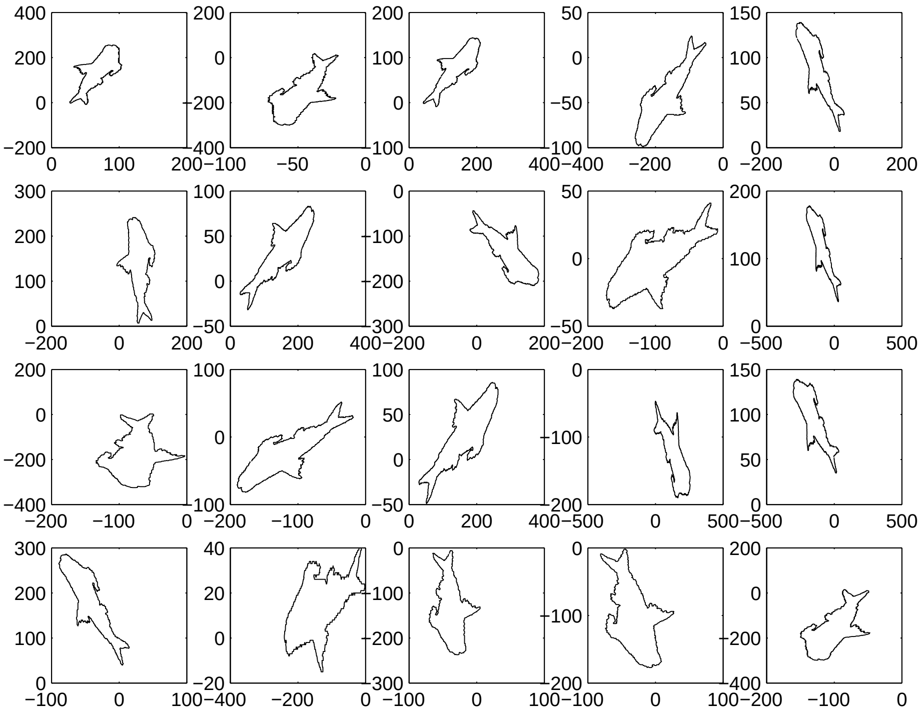


Figure 3.3 Variations of fish contours for type 1 in database. Contour deformations are generated by applying affine transformations on a fish prototype. Twenty deformations are produced for each prototype.

3.3.2 Experimental Setup and Results

A fice-way cross validation is conducted to evaluate the shape matching performance of the summation invariants. The details of the experimental procedures are described below:

1. A subset of 100 prototypes of marine animals are selected from the SQUID dataset [37]. For each prototype, 20 deformations are generated by applying affine transformations on the prototype. Hence, a dataset containing 2000 shapes of marine animals is created.

2. The dataset is divided into 5 equal-sized partitions. In each partition, there are 100 prototypes and 4 deformations associated with each prototype. In other words, each partition will contain 400 shapes that correspond to four variations of each of the 100 fish prototypes.

3. In one shape matching experiment, one partition will be used as *gallery* set and the others will be used as *probe* set. Since the gallery set and the probe set are mutually exclusive, one could evaluate how well a shape matching algorithm could be generalized to shapes it has never seen before. More specifically, a probe shape will be matched against all of the gallery shapes. Then, we can pick up the pair of shapes with highest matching score. If these two shapes belong to the same prototype, we treat it as a correct matching result. If these two shapes do not belong to the same prototype, it is regarded as an incorrect match. In my experiments, the matching score, *a.k.a.* similarity score, between two shapes is given by:

$$\rho = \frac{\sum_{n=0}^{N-1} v_{gallery}[n] \cdot v_{probe}[n]}{\sqrt{\sum_{n=0}^{N-1} v_{gallery}^2[n] \cdot \sum_{n=0}^{N-1} v_{probe}^2[n]}} \tag{3.3}$$

4. In a M-way cross-validation method ($M = 5$ in our experiment), one can perform M experiments. In the first experiment, the first partition will be used as gallery set and the others will be used as probe sets. Similarly, in the second experiment, the second partition will be used as gallery set and the others will be used as probe sets. Therefore, given M partitions of the whole dataset, one could conduct a total of M experiments. This way, each partition will be used as the gallery set exactly once in a M-way cross-validation.

5. For each experiment, the shape matching results are reported by a 100×100 confusion matrix $\mathbf{C}$. c_{ij} denotes the $(i, j)^{th}$ element of $\mathbf{C}$ and is the number of probe shapes that actually belong to prototype i, but are considered as prototype j by a shape matching algorithm. The confusion matrices of all M shape matching experiments will be added together to form the final confusion matrix, from which one can perform different kinds of statistical analysis. For example, the probability of incorrect-matching can be calculated as the ratio of the sum of all the off-diagonal elements to the sum of all the elements.

The shapes of the marine animals in the SQUID dataset are simple and are closed contours, *i.e.* no self intersection. For matching two closed contours, one needs to first align two contours and then compute their similarity score. In this experiment, the shape matching procedure is as follows:

1. Circularly shift the feature vector from the probe set by one and compute the corresponding similarity score ρ with the one from the gallery set.

2. Repeat the previous step N times, where N is the number of points on a probe shape. Choose the maximum as the final similarity score between two shapes.

Using semi-local summation invariant, the five-way cross validation produces 177 matching errors. Therefore, the probability of incorrect-matching is

$$\wp_{summation}(\text{error}) = \frac{177}{8000} = 2.21\%$$

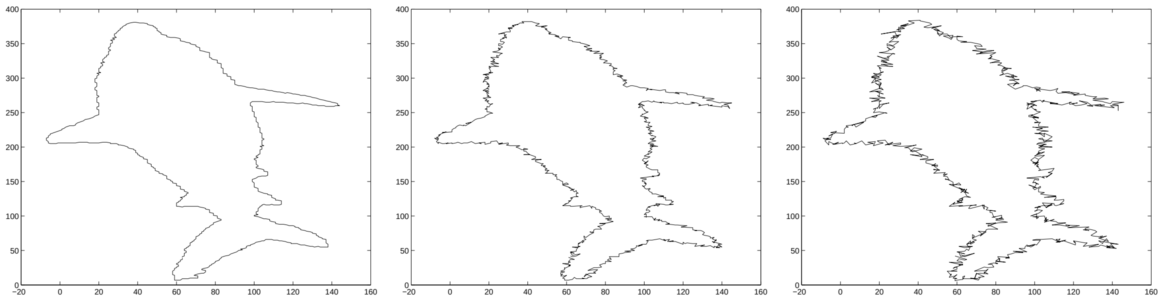
Using *semi-local integral invariant*, the probability of incorrect-matching is

$$\wp_{integral}(\text{error}) = \frac{1165}{8000} = 14.56\%$$

Using *wavelet-based invariant*, the probability of incorrect-matching is

$$\wp_{wavelet}(\text{error}) = \frac{696}{8000} = 8.7\%$$

Obviously, the summation invariant shows better discriminating capability than that of the wavelet-based invariant and the integral invariant.



(a) σ = 0

(b) σ = 1

(c) σ = 2

Figure 3.4 Fish contours before and after adding Gaussian distributed noise.(σ denotes the standard deviation of noise)

Table 3.1 Error Rate of Fish Recognition at Different Noise Level

	Summation Invariant	Integral Invariant	Wavelet Invariant
$\sigma = 0$	$\frac{177}{8000} = 2.21\%$	$\frac{1165}{8000} = 14.56\%$	$\frac{696}{8000} = 8.7\%$
$\sigma = 1$	$\frac{367}{8000} = 4.59\%$	$\frac{3726}{8000} = 46.58\%$	$\frac{2253}{8000} = 28.16\%$
$\sigma = 2$	$\frac{514}{8000} = 6.43\%$	$\frac{6012}{8000} = 75.15\%$	$\frac{3476}{8000} = 43.45\%$

σ denotes the standard deviation of Gaussian distributed noise.

3.3.3 Sensitivity to Noise

In this experiment, the perturbation caused by noise will be investigated. We add Gaussian noise into the fish contours and then quantize the coordinates of fish contours. Fish contours with different noise levels are shown in Figure 3.3.3. For each fish contour, the *semi-local summation invariant*, *semi-local integral invariant* and *wavelet-based invariant* are computed. Once again, five-way cross validation is utilized to evaluate the shape matching performances of these three invariant features. The probabilities of incorrect-matching are summarized in Table 3.1. Evidently, the summation invariant exhibits better noise immunity than the other two methods.

3.4 Nonrigid Matching Using Summation Invariants

Shape analysis is a vast area and it plays an important role in systems for object recognition, matching and registration. Some excellent reviews on this topic are available in [38, 39]. Here, we will discuss the issue of shape alignment, which is a sub-area in shape analysis. Basically, the goal of shape alignment is to establish a correspondence between two shapes. Many of the alignment algorithms work in the following way. Let s_1 be the arc-length of curve C, s_2 be the arc-length of curve $\tilde{C}$ and ϕ be a mapping from s_1 to s_2. Of all possible mappings, the one which minimizes the cost function is considered as the optimal alignment between C and $\tilde{C}$. The problem can be formulated as shown below.

$$\arg\min_{\phi} \int_C \left\{\kappa_1(s_1) - \kappa_2(\phi(s_1))\right\}^2 ds_1 \tag{3.4}$$

where κ_1 and κ_2 are curvatures of C and $\tilde{C}$ respectively. This formulation is simple and indeed captures the fundamental issue behind aligning two shapes. However, its solution is asymmetric. Since the cost function is integrated along s_1, changing the labels of the two curves might result in different optimal alignments. To see this, consider the following equation which changes the variable of integration from s_1 to s_2.

$$\begin{aligned} & \int_C \left\{\kappa_1(s_1) - \kappa_2(\phi(s_1))\right\}^2 ds_1 \\ = \ & \int_{\tilde{C}} \left\{\kappa_1(\phi^{-1}(s_2)) - \kappa_2(s_2))\right\}^2 \frac{d\phi^{-1}(s_2)}{ds_2} ds_2 \end{aligned}$$

This says that the optimal alignment ϕ^{-1} can be found by minimizing the cost function below.

$$\int_{\tilde{C}} \left\{\kappa_1(\phi^{-1}(s_2)) - \kappa_2(s_2))\right\}^2 w(s_2) ds_2$$

Due to the existence of the weight $w(s_2)$, the mapping ϕ^{-1} does not necessarily minimize the cost function $\int_{\tilde{C}} \left\{\kappa_1(\phi^{-1}(s_2)) - \kappa_2(s_2))\right\}^2 ds_2$.

3.4.1 Symmetric Problem Formulation

The asymmetry issue can be eliminated by using the alignment curve [40], resulting in a revised problem formulation. The alignment between two curves C and $\tilde{C}$ is specified in terms of two functions h_1 and h_2. The functions h_1 and h_2 share a common domain denoted by ξ. The function h_1 is a mapping from ξ to s_1. Similarly, the function h_2 is a mapping from ξ to s_2. Both h_1 and h_2 are not necessary invertible. This assumption makes alignment between the two curves more flexible. In particular, this allows the cases where a correspondence is many-to-one or one-to-many. When h_1 is invertible, one can establish a direct mapping from s_1 to s_2, *i.e.* $s_2 = h_2(h_1^{-1}(s_1)) = h_2 \circ h_1^{-1}(s_1)$. Similarly, one has a direct mapping from s_2 to s_1 when h_2 is invertible. The case where both h_1 and h_2 are invertible does not exist. This will soon be clear when we introduce the notation of alignment curve.

By introducing the shared parameter ξ, the original problem formulation can be transformed into a symmetric one. More specifically, the cost function is revised so that it is an integral over ξ.

Now, we will move on to the details of the symmetric formulation. Let α be the alignment curve with coordinates $(h_1(\xi), h_2(\xi))$.

$$\alpha(\xi) \triangleq (h_1(\xi), h_2(\xi)), \quad \xi \in [0, \tilde{L}] \tag{3.5}$$

$$\alpha(0) = (0, 0) \tag{3.6}$$

$$\alpha(\tilde{L}) = (L_1, L_2) \tag{3.7}$$

where ξ is arc-length along the alignment curve. L_1, L_2 and $\tilde{L}$ are the total length along C, $\tilde{C}$ and the alignment curve α, respectively. The angle between the tangent to the alignment curve and the x-axis is denoted by ψ. An example of an alignment curve is shown in Figure (3.5). The arc-length parameters s_1 and s_2 can then be expressed in terms of ψ, as shown below.

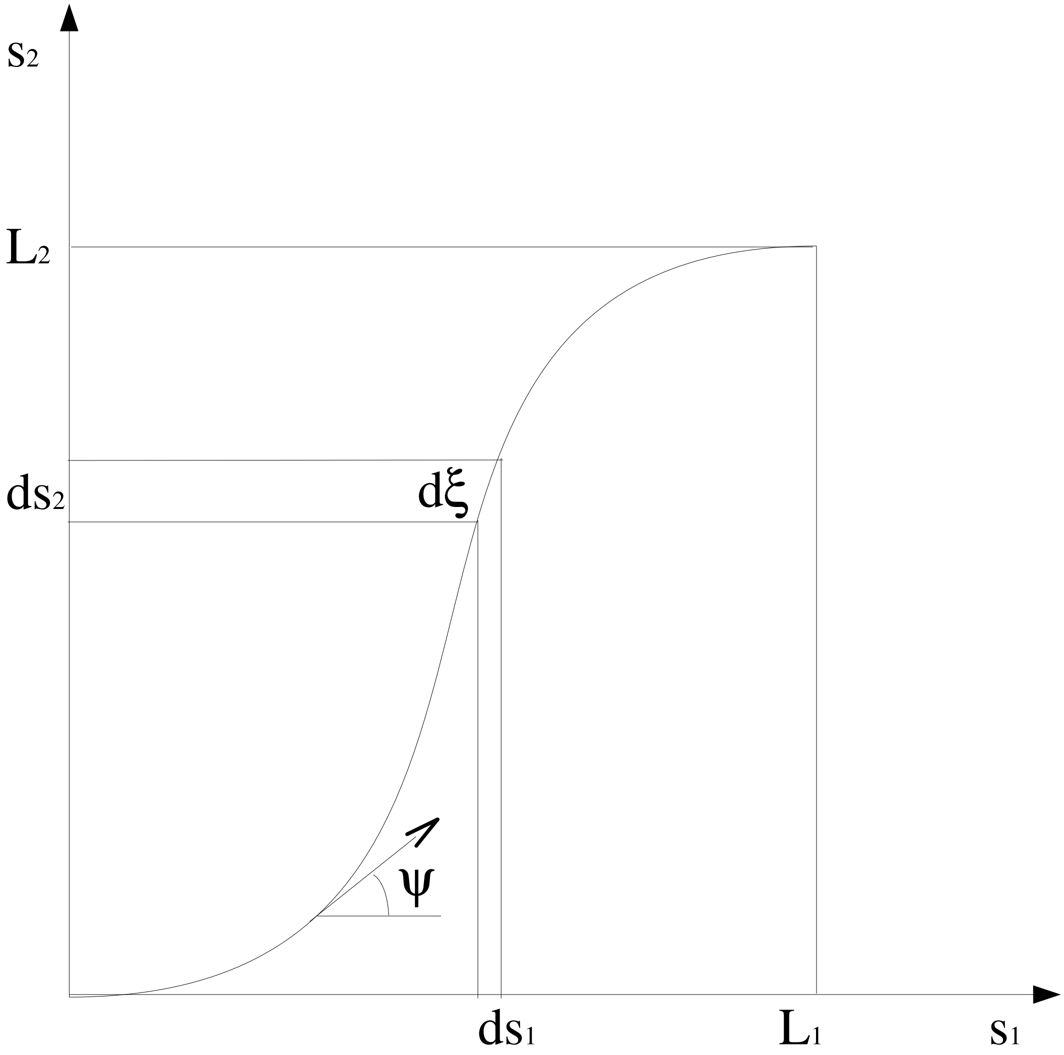


Figure 3.5 By introducing the alignment curve α in the problem formulation, the optimal solution allows for a segment from one curve corresponding to a single point on the other curve, *i.e.* many-to-one correspondence, as well as a single point from one curve corresponding to a segment in the other curve, *i.e.* one-to-many correspondence.

$$s_1(\xi) = \int_0^\xi \cos(\psi(\tau))d\tau \tag{3.8}$$

$$s_2(\xi) = \int_0^\xi \sin(\psi(\tau))d\tau \tag{3.9}$$

Note that an alignment should preserve the ordering of points on curves C and $\tilde{C}$, *i.e.* if $\xi_B > \xi_A$, then $h_1(\xi_B) \geq h_1(\xi_A)$ and $h_2(\xi_B) \geq h_2(\xi_A)$. Hence, ψ is constrained by the order preserving property to lie in $[0, \frac{\pi}{2}]$. The cost function of the matching between C and $\tilde{C}$ can now be re-written in terms of ψ.

Consider an infinitesimal curve segment, shown in Figure (3.6), with the initial point A and the end point B. Its arc-length is denoted by ds. The tangent vectors at point A and point B are denoted by $\vec{T_A}$ and $\vec{T_B}$ respectively. The angle between $\vec{T_A}$ and $\vec{T_B}$ is denoted by $d\theta$.

$$\cos(d\theta) = \frac{< \vec{T_A}, \vec{T_B} >}{\|\vec{T_A}\| \cdot \|\vec{T_B}\|} \tag{3.10}$$

Given two infinitesimal curve segments $C|_{[h_1(\xi_a), h_1(\xi_b)]}$ and $\tilde{C}|_{[h_2(\xi_a), h_2(\xi_b)]}$, the cost of matching can be formulated as

$$|ds_1 - ds_2| + R|d\theta_1 - d\theta_2| \tag{3.11}$$

where R is a regularization term. In this formulation, the first term penalizes *stretching* and the second term penalizes *bending*. It can be expressed in terms of ψ by changing variables, as shown below.

$$
\begin{aligned}
& |ds_1 - ds_2| + R|d\theta_1 - d\theta_2| \\
= \quad & \Big(|\frac{ds_1}{d\xi} - \frac{ds_2}{d\xi}| + R|\frac{d\theta_1}{d\xi} - \frac{d\theta_2}{d\xi}|\Big)d\xi \\
= \quad & \Big(|\frac{ds_1}{d\xi} - \frac{ds_2}{d\xi}| + R|\frac{d\theta_1}{ds_1}\frac{ds_1}{d\xi} - \frac{d\theta_2}{ds_2}\frac{ds_2}{d\xi}|\Big)d\xi \\
= \quad & \Big(|\frac{dh_1}{d\xi} - \frac{dh_2}{d\xi}| + R|\frac{d\theta_1}{ds_1}\frac{dh_1}{d\xi} - \frac{d\theta_2}{ds_2}\frac{dh_2}{d\xi}|\Big)d\xi \\
= \quad & \Big(|\cos(\psi) - \sin(\psi)| + R|\frac{d\theta_1}{ds_1}\cos(\psi) - \frac{d\theta_2}{ds_2}\sin(\psi)|\Big)d\xi \\
= \quad & \Big(|\cos(\psi) - \sin(\psi)| + R|\kappa_1(h_1(\xi))\cos(\psi) - \kappa_2(h_2(\xi))\sin(\psi)|\Big)d\xi \qquad (3.12)
\end{aligned}
$$

The optimal alignment α is the one which satisfies the following criterion.

$$\arg\min_{\alpha} \int_0^{\tilde{L}} |\cos(\psi) - \sin(\psi)| + R|\kappa_1(h_1(\xi))\cos(\psi) - \kappa_2(h_2(\xi))\sin(\psi)|d\xi \qquad (3.13)$$

subject to

$$\begin{cases} 0 \leq \psi \leq \frac{\pi}{2} \\ \alpha(0) = (0,0) \\ \alpha(\tilde{L}) = (L_1, L_2) \end{cases}$$

So far, the problem formulation is still based on the differential invariant, *i.e.* curvature. However, some summation invariants have the same geometric properties as curvature and therefore can replace curvatures in Equation (3.13). For example, $\eta_{0,1}$ is invariant to rigid transformation and yields high values at high curvature regions. By replacing curvature with $\eta_{0,1}$, the second term in Equation (3.13) still has its functionality, *i.e.* penalizing bending between matched segments. Hence, the cost of matching two shapes can be defined as shown below.

$$\int_0 \tilde{L}|\cos(\psi) - \sin(\psi)| + R|\eta^1_{0,1}(h_1(\xi))\cos(\psi) - \eta^2_{0,1}(h_2(\xi))\sin(\psi)|d\xi \tag{3.14}$$

where $\eta^1_{0,1}$ and $\eta^2_{0,1}$ are Euclidean summation invariants of C and $\tilde{C}$, respectively.

3.4.2 Finding the Optimal Alignment Curve by Dynamic Programming

The minimization problem formulated in the previous section can be solved efficiently using dynamic programming. There are many shape matching methods based on the dynamic programming algorithm [28, 29, 41]. In this section, we will describe a dynamic-programming algorithm for finding the optimal alignment curve.

Let C and $\tilde{C}$ be the two curves to be matched. $s_1, s_2, \ldots, s_n$ and $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_m$ are sample points on C and $\tilde{C}$ respectively. The alignment curve is then represented by a sequence of N points $\xi_1, \xi_2, \ldots, \xi_N$ where

$$\begin{cases} \xi_k = (s_{i_k}, \tilde{s}_{j_k}), \quad i_k \in \{1, \ldots, n\}, j_k \in \{1, \ldots, m\}, k \in \{2, \ldots, N-1\} \\ \xi_1 = (s_1, \tilde{s}_1) \\ \xi_N = (s_n, \tilde{s}_m) \end{cases}$$

A table of cumulative cost is built and the optimal alignment is searched in the form of a path in the dynamic-programming table. The DP table has n columns and m rows. The index on column is denoted i which corresponds to sample points on C. Similarly, the index on row is denoted j which corresponds to sample points on $\tilde{C}$. An example of a DP table with 16 columns and 14 rows is shown in Figure (3.7).

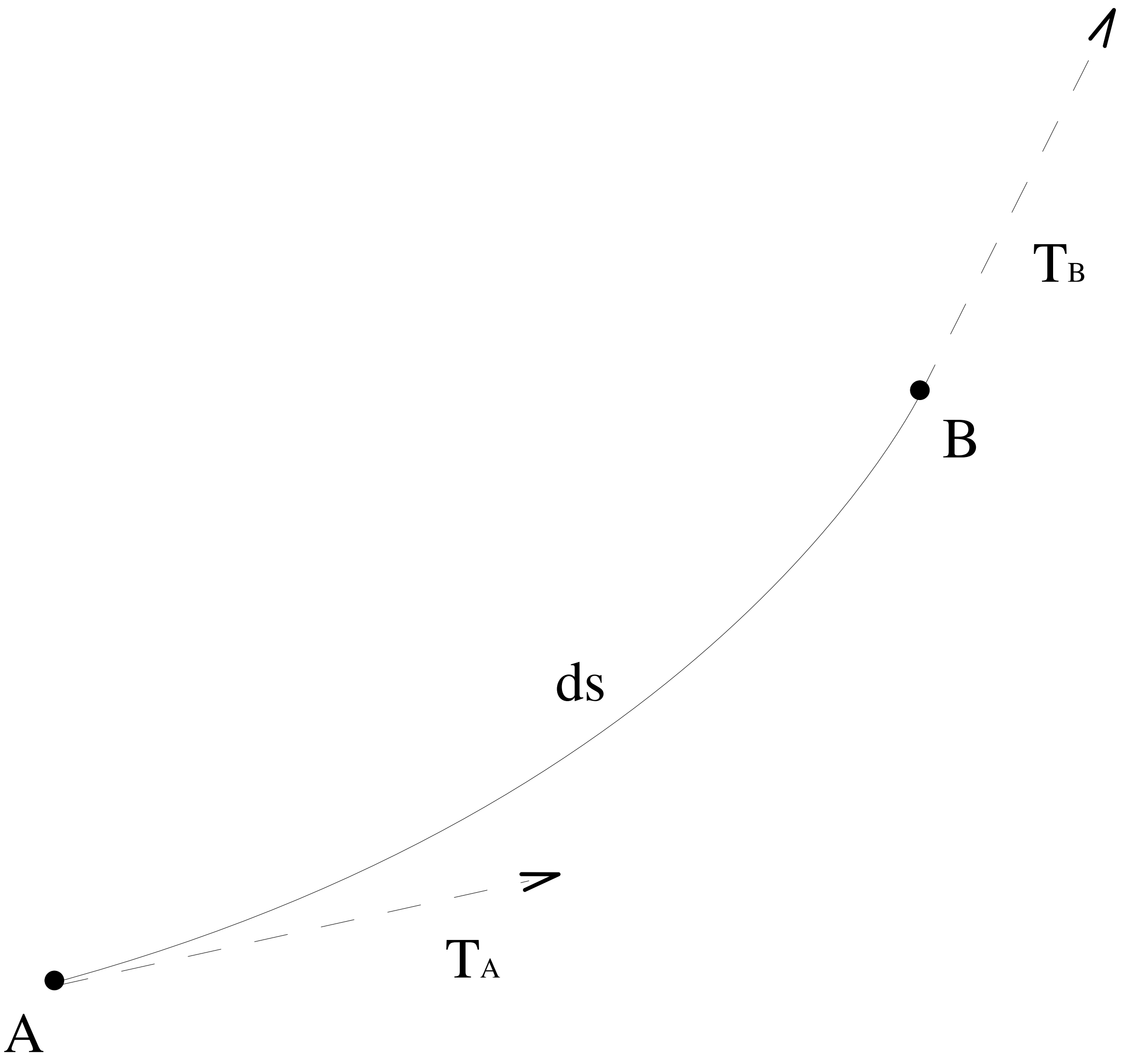


Figure 3.6 An infinitesimal curve segment with arc-length ds. The tangent vectors at point A and point B are denoted by T_A and T_B respectively.

A link between the entries (i_{k-1}, j_{k-1}) and (i_k, j_k) denote the matching of segments $C|_{[i_{k-1},i_k]}$ and $\tilde{C}|_{[j_{k-1},j_k]}$. Let $d(i,j)$ denote the cost of matching the curve segments $C|_{[s_1,s_i]}$ and $\tilde{C}|_{[\tilde{s}_1,\tilde{s}_j]}$, and let $\delta([k,i],[\ell,j])$ denote the cost of matching segments $C|_{[s_k,s_i]}$ and $\tilde{C}|_{[\tilde{s}_\ell,\tilde{s}_j]}$. The DP table is updated by the following rule:

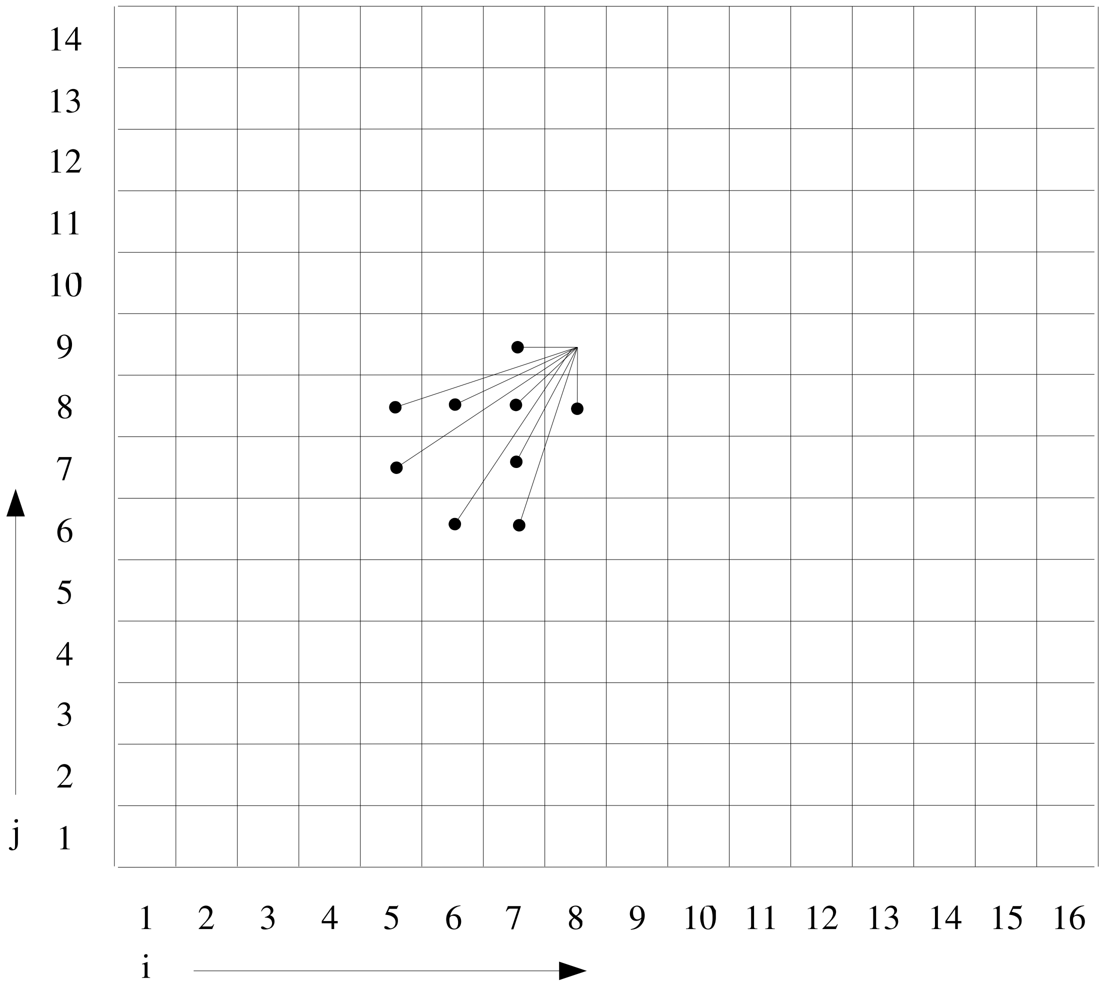


Figure 3.7 An example of DP table. Discrete sample points along C and $\tilde{C}$ are the x-axis and y-axis respectively. The value of $d(i,j)$, stored at the entry (i,j), is updated by considering the 9 possible paths shown here.

$$d(i,j) = \min_{k,\ell} \left\{ d(i-k, j-\ell) + \delta([i-k,i],[j-\ell,j]) \right\} \tag{3.15}$$

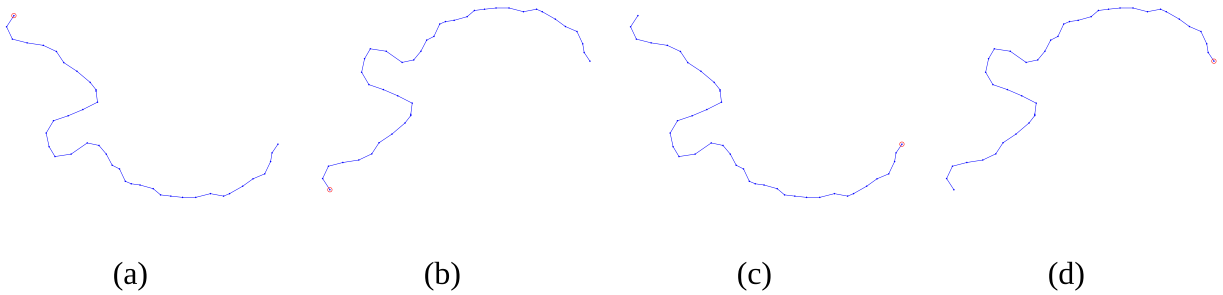


Figure 3.8 4 cases to be considered during shape matching. (a) The original shape, (b) The symmetric shape, (c) The original shape with opposite starting point, and (c) The symmetric shape with opposite starting point. The starting point on each curve is denoted by a red dot.

The parameter ψ is quantized during the computation of the partial cost. In Figure 3.7, one can see the 9 possible local paths which correspond to the 9 quantized values of ψ.

$$\psi \in \{0\,^\circ, 18\,^\circ, 27\,^\circ, 34\,^\circ, 45\,^\circ, 56\,^\circ, 63\,^\circ, 72\,^\circ, 90\,^\circ\}. \tag{3.16}$$

Note that $\psi = 0\,^\circ$ serves the purpose of deleting a curve segment on $\tilde{C}$, and $\psi = 90\,^\circ$ corresponds to inserting a curve segment on $\tilde{C}$. One has to sequentially update each entry $d(i, j)$ in the DP table. Finally, the optimal alignment is found by tracing through the DP table. The complexity of dynamic-programming is $O(nm)$, where n, m are the number of sample points on C and $\tilde{C}$, respectively.

A shape matching algorithm should be capable of handling symmetric shapes and different starting points. Figure 3.8 illustrates all these cases for an open contour; C_1 is the original shape, C_2 is a symmetric shape, C_3 has a different starting point (opposite traversal), and C_4 is the combination of C_2 and C_3. These 4 possible cases should be considered during shape matching. The same argument holds for matching closed shapes. The cases with symmetric shapes or different starting point are handled by repeating the DP algorithm on each possible case. Each case will produce a matching cost and the minimum is taken as shape distance.

3.4.3 Experimental Results

Several experiments are conducted by using open and closed shapes. The following dataset is used in the experiments.

SQUID [37] There are 1100 shapes of marine animals. These shapes are simple and closed contours, *i.e.* no self intersection. Twenty out of 1100 shapes are selected as queries. Also, 1500 open shapes are generated from the closed shapes by manually editing, *i.e.* deleting about half of each shape.

Each shape is represented by a sequence of points. All shapes are preprocessed by Petrakis *et. al.* [1] so that their lengths range from 80 to 100. These datasets, including open shapes, are available on the Internet, http://www.ced.tuc.gr/∼petrakis.

Figure 3.9 shows the results of open shape retrieval. The results reported by Petrakis *et. al.*'s algorithm [1] are shown in Figure 3.10. In both figures, the query shape is located in the first row and the retrieved 20 best matches are placed in the following rows. The shapes retrieved by the two algorithms are very similar. This demonstrates that the proposed algorithm achieves at comparable performance with the state-of-the-art. However, differences can still be observed by taking a closer look at the retrieved shape, Target 702, in Figure 3.10. Obviously, it is an incorrect matching. On the other hand, there is no such obvious mistake in Figure 3.9.

Figure 3.11 and 3.12 show the results of closed shape retrieval. The proposed algorithm yields many incorrect matches in this case, *e.g.* target 398 and 140 in the second row of Figure 3.11. One possible explanation is that the window size for computing semi-local summation invariants is not chosen properly. It has a direct impact on the resulting shape representation. On one hand, a larger window is more robust to noise but also has more blurring effect. On the other hand, a smaller window has less blurring effect but is less robust to noise. Hence, a trade off has to be made which is a highly application-specific decision.

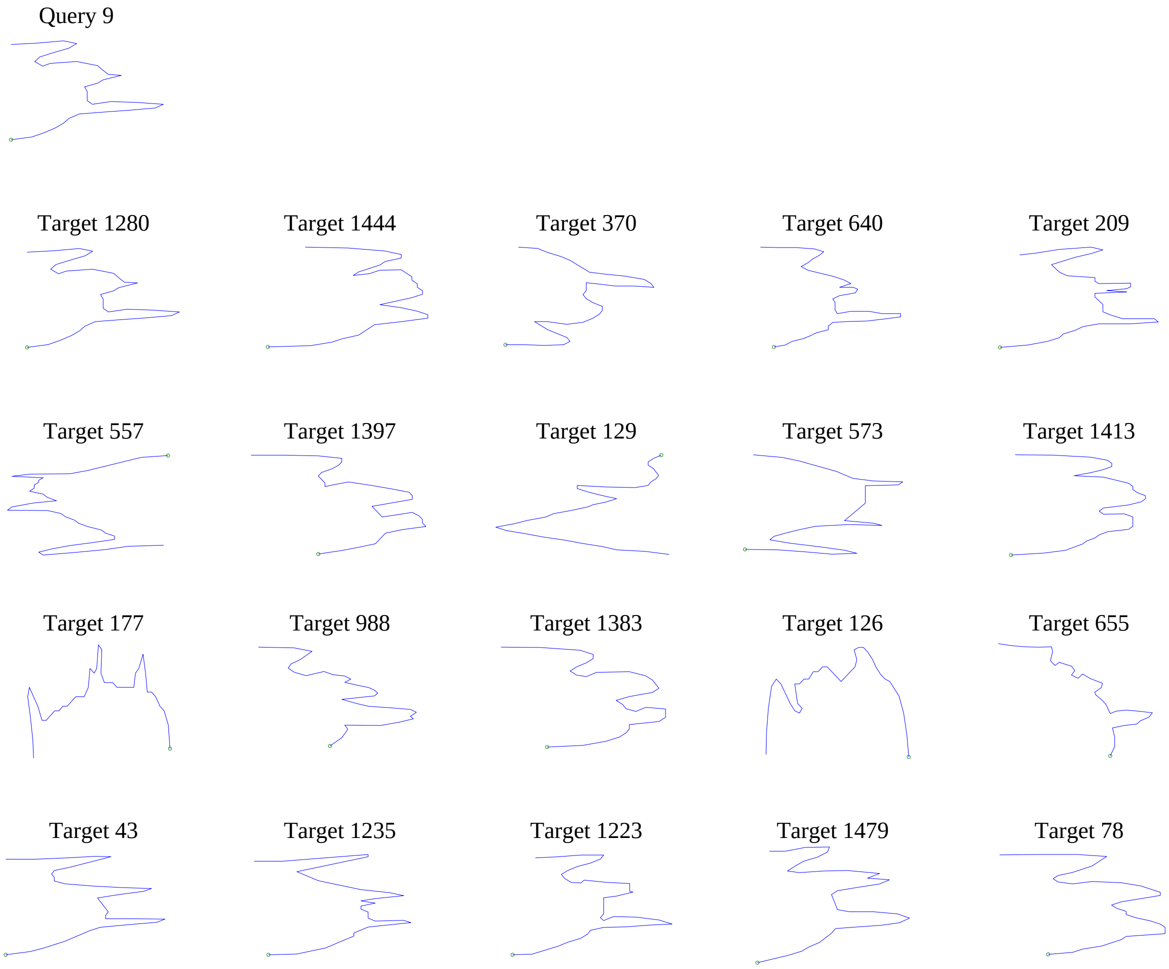


Figure 3.9 An open query and its 20 best matches retrieved by the proposed shape matching algorithm.

3.5 Summary

The feasibility of applying summation invariants in 2D shape analysis is discussed in this chapter. A novel shape representation based on summation invariants is introduced. In the form of semi-local summation invariant representation, a planar curve is represented by the values of semi-locally computed summation invariants. It could be tailored so that it is only invariant to a particular type of geometric transformations, *i.e.* discrimination capability is maximized. Also, the

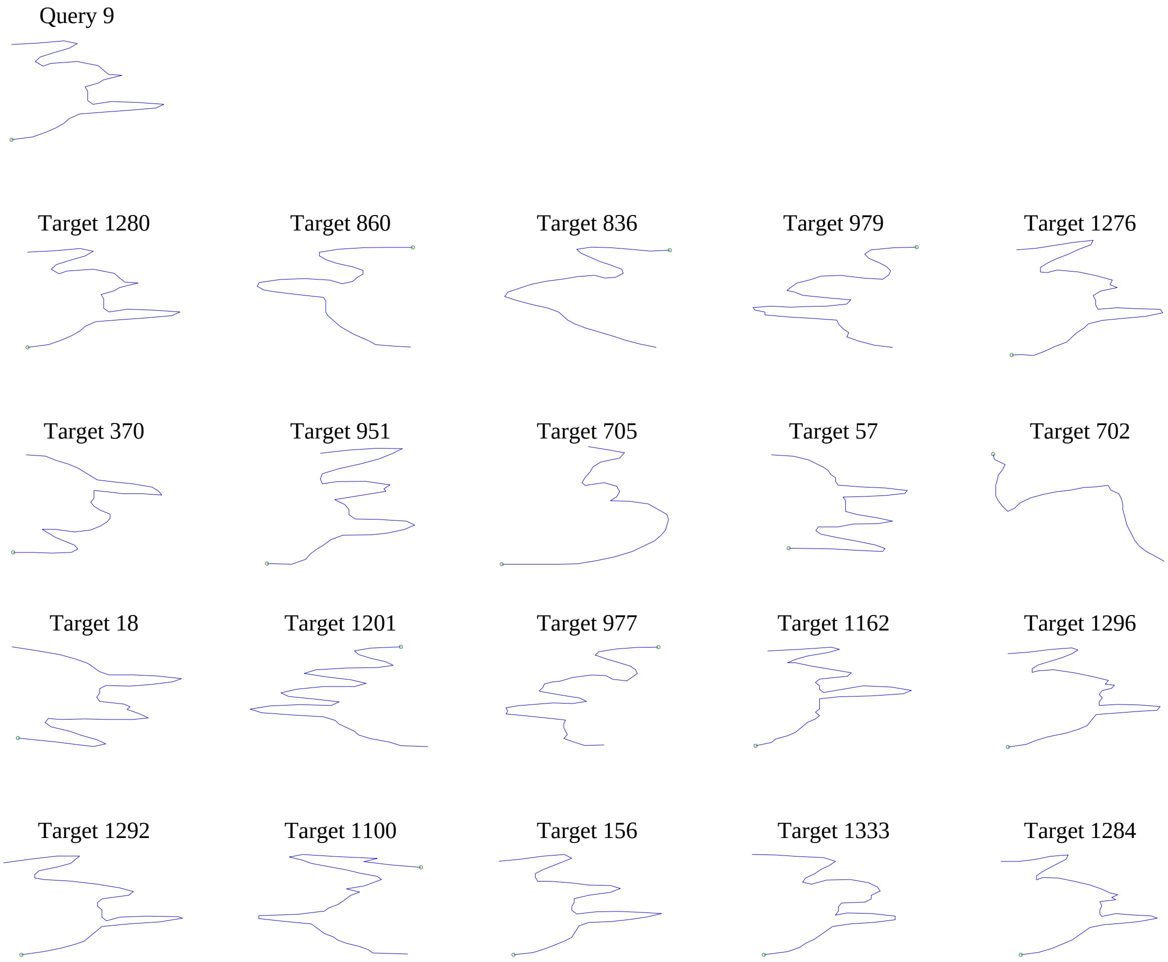


Figure 3.10 An open query and its 20 best matches retrieved by Petrakis *et. al.*'s algorithm [1].

geometric characteristics of a planar curve are well preserved by the proposed shape representation and hence it is useful for general shape analysis problems.

Two sets of experiments are conducted to evaluate the performance of using the proposed shape representation. In the first set of experiments, the proposed representation exhibits better discriminative capability and better noise immunity than some conventional shape representations.

In the second set of experiments, a novel shape matching algorithm which combines dynamic programming and the proposed shape representation is tested on the dataset of marine animals. It successfully performs matching and retrieval on shapes with considerable amounts of deformation

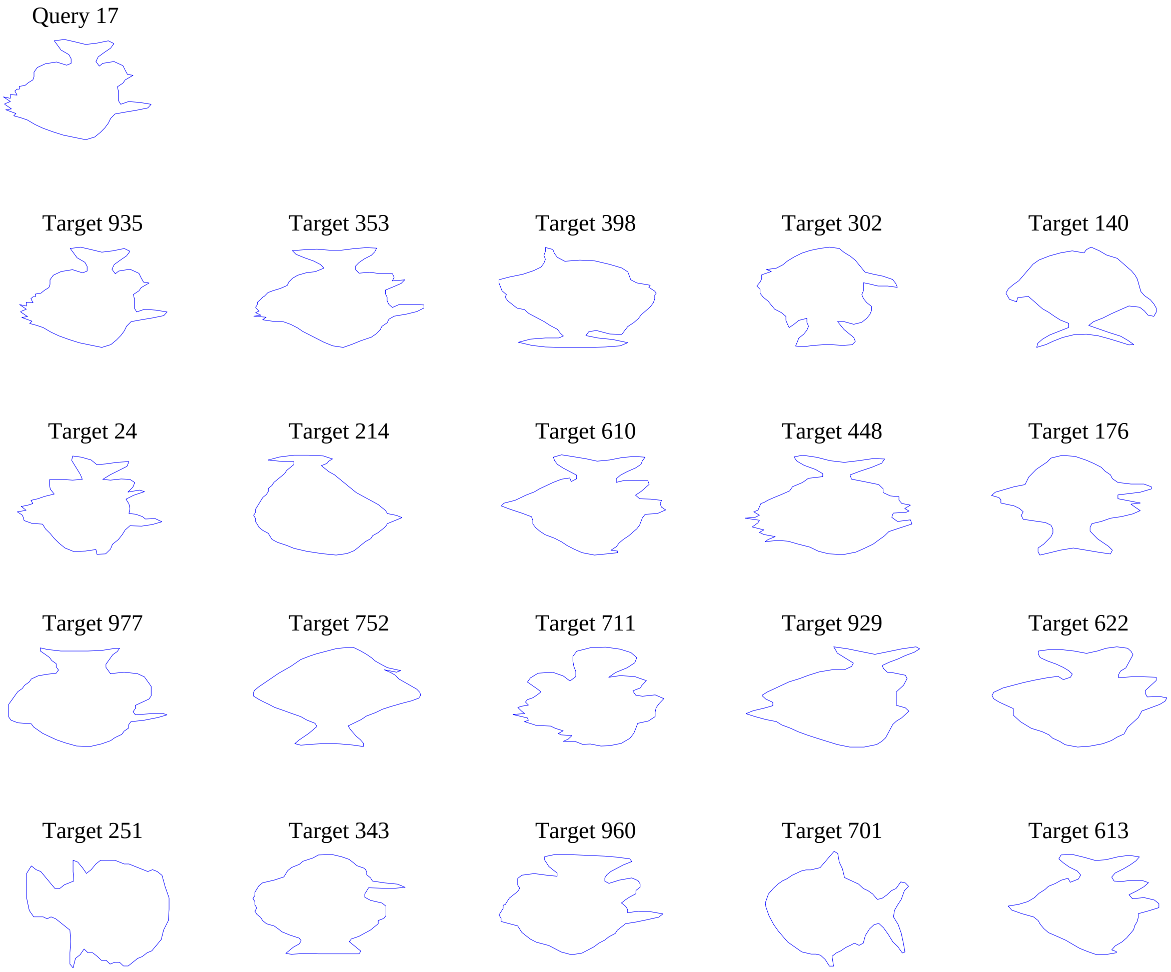


Figure 3.11 An closed query and its 20 best matches retrieved by the proposed shape matching algorithm.

and achieves comparable performance with the-state-of-the art algorithms [1]. The proposed shape matching algorithm is capable of handling both open and closed shapes.

Among the possible future directions, more comprehensive experiments on shape retrieval are necessary. However, human judgment is inevitably involved in evaluating the results of shape retrieval. In order to have statistically meaningful results, a suitable number of human subjects are required.

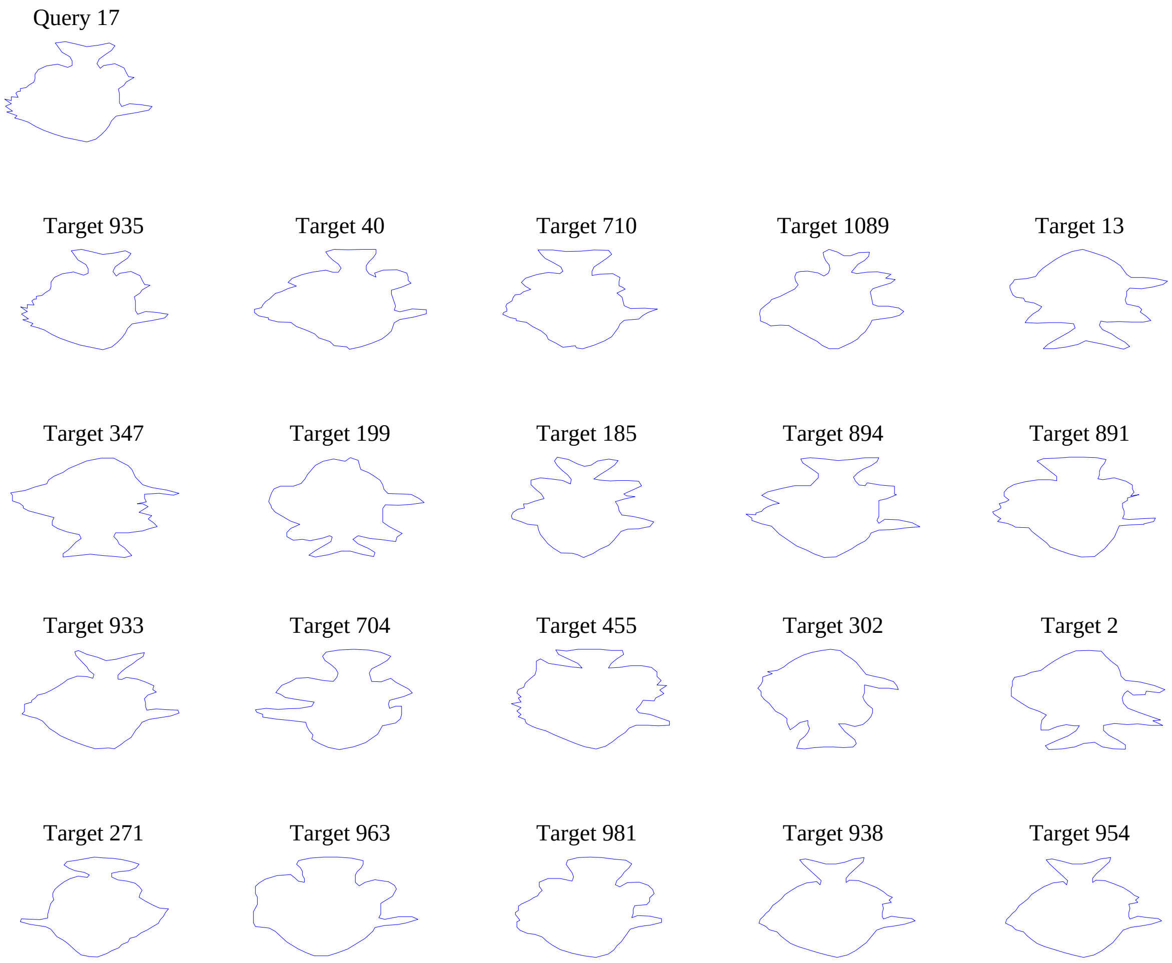


Figure 3.12 An closed query and its 20 best matches retrieved by Petrakis *et. al.*'s algorithm [1].

Chapter 4

3D Face Recognition

4.1 Introduction

Face recognition technology has many advantages that make it a preferable choice for biometric authentication. In fact, it is one of the fastest growing biometric technologies because cooperation from the user is NOT required. Face recognition can be deployed in high volume applications where obtaining other biometrics, such as fingerprint or iris, is impractical. However, rigorous tests with real-world data, such as FERET and FRVT, have revealed many shortcomings of existing approaches [42, 43]. In particular, for large scale, real world situations, current algorithms still cannot meet the performance requirement for practical applications.

A majority of face recognition approaches make use of typical texture images which are subject to illumination, pose, and age (temporal) variations, and can easily be altered through simple make-up efforts. On the other hand, if 3D facial surfaces are available, one may exploit features that are invariant to appearance variations. For example, the facial surface around cheek bones or the nose would remain unchanged under varying lighting conditions, are less likely to change due to aging, and are seldom covered with hairs.

With 3D facial surfaces, the issue of illumination variation can be mostly alleviated; and the pose variation problem may be normalized successfully [44] if the location of facial features, such as nose tip, corners of eyes, ..., etc, are given. Unfortunately, one issue remains due to the non-rigid nature of the human face. So far, performing face recognition in the presence of facial expressions is still one of the most challenging problems in pattern recognition.

In the following sections, we will introduce the terminologies and briefly review some techniques used in face recognition.

4.1.1 Technical Terminology

The term **recognition** is not well defined and can refer to two different scenarios. One scenario is called **identification** and the other is called **verification**. For both scenarios, a system will match a probe image against a gallery image. The **probe** set means a collection of images of various people who are NOT known to the system. In contrast, **gallery** set means a collection of images of various people who are known to the system. In an identification scenario, a user does not claim his/her identity. Hence, a system will search the whole database for the closest match (a.k.a. one-to-many matching). In a verification scenario, a user will claim his/her identity and matching is performed against his/her template (a.k.a. one-to-one matching). The identification scenario is more challenging because matching against a set of images tends to produce more incorrect results.

The performance of a recognition system can be reported in different ways. On way of reporting performance is through a **Cumulative Match Characteristics** (CMC) curve. A CMC curve shows, in an explicit way, the trade off between true positive rate versus false positive rate. In a CMC curve, the y axis is the true positive rate and the x axis is the **cumulative rank**, which is the maximum possible matches that a system is allowed to report. If a system is allowed to report a large number of possible matches, the true positive rate will increase accordingly. However, the false positive rate will also increase, as will the work load on human operators. In other words, performance gain is at the price of increasing the burden on humans.

System performance can also be described by a **Receiver Operating Characteristics** (ROC) curve. It is a plot of the true positive rate against the false positive rate for the different possible threshold values. The **Equal Error Rate** (EER), the point where false positive rate equals to false negative rate, is also commonly used in reporting performance. The lower the EER value, the higher the accuracy of the biometric system.

4.1.2 Representation Techniques

Principal Component Analysis (PCA) is widely used as a dimensionality reduction procedure. Specifically, PCA computes an orthogonal basis that leads to a low dimensional representation of an input image. Turk and Pentland [45] introduced the *Eigenface*, where the eigenfaces are the eigenvectors associated with the largest eigenvalues. The eigenfaces span a lower dimensional space. Performing classification in the reduced space simplifies this task.

The Gabor wavelets provide desirable properties such as spatial locality and orientation selectivity [46]. The biological relevance and computational properties of Gabor wavelets have been discussed in [47–49]. Lades *et al.* [50] introduced the dynamic link architecture (DLA) for face recognition based on Gabor wavelets representation. Liu and Wechsler [51] proposed a Gabor-Fisher Classifier (GFC) which is robust to illumination variation and facial expressions. The GFC method combines the Fisher linear discriminant model and Gabor representation of face images. A recent study [52] indicates that Gabor representation provides better performance than some other techniques.

4.1.3 Classification Techniques

The Bayes classifier provides the optimal performance if the probability density function is known. One can improve the PCA performance by integrating PCA with the Bayes classifier. The Probabilistic Reasoning Models (PRM) [53] method first applies PCA for dimensionality reduction and then estimates the probability density functions for each class. Moghaddam *et al.* [55] proposed a similarity measure for image matching based on Bayesian analysis.

Linear Discriminant Analysis (LDA), or Fisher Linear Discriminant (FLD), is a widely used approach in pattern classification. Basically, LDA yields the basis which best separates two or more

Table 4.1 Representation Techniques and Their Applications to Face Recognition

Representation techniques	Face recognition algorithms
PCA	Eigenfaces [45], PRM [53]
Gabor wavelets	DLA [50], EBGM [54], GFC [51]
Bayes classifier	PRM [53], Bayesian analysis [55]
LDA/FLD	Fisherfaces [56], LDA [57], EFC [58]
ICA	ICA [59], EICA [60]
Graph matching	DLA [50], EBGM [54]

classes of object. LDA is also related to PCA in that both look for linear combinations of a basis which best explain the data. However, LDA explicitly attempts to model the difference between the classes, while PCA does not take into account any difference in class. Many face recognition algorithms have been proposed to improve discriminating capability [56–58] by using LDA. The Fisherfaces [56] method first finds a lower dimensional space, where features are obtained by LDA criterion for face recognition. Note that LDA-based approaches yield better performance only when training data provides a good representation of image variations. Otherwise, the performance difference between LDA and PCA is not significant.

Independent Component Analysis (ICA) minimizes both second-order and higher-order dependencies and attempts to find the basis along which the data (when projected onto them) are statistically independent. Bartlett *et al.* [59] provided two ICA architectures for face recognition task: Architecture I - statistically independent basis images, and Architecture II - factorial code representation. Liu and Wechsler [60] presented the Enhanced ICA (EICA) and its application in face recognition.

Graph-based methods have the advantage for performing face recognition in the presence of facial expressions. Lades *et al.* [50] proposed the dynamic link architecture (DLA) for face recognition. Wiskott *et al.* [54] introduced Elastic Bunch Graph Matching (EBGM) which is an extension of the DLA method. Faces are represented by *labeled graphs*, based on a Gabor wavelet representation of face images. Image graphs of new faces are extracted by an elastic graph matching process and can be compared using a simple similarity function.

4.1.4 Previous Work in 3D Face Recognition

The majority of face recognition research focuses on using intensity images of the face. However, 3D images of the face have several advantages over the intensity-based features. In particular, 3D data provide a better representation for describing properties of the face in areas such as the cheeks, forehead, and chin, and are illumination invariant.

A comprehensive survey on 3D face recognition can be found in [61]. Here, we focus on those involving invariant features or dealing with the issue of facial expressions.

The early work of applying invariant functions to 3D face recognition was done over a decade ago. At that time, people began with the geometrical properties introduced in differential geometry, such as curvature. Cartoux *et al.* [62] proposed a face recognition algorithm based on Gaussian curvature of the face surface. Their approach yields a 100% verification rate on a small dataset (5 subjects and 18 range images). Lee and Milios [63] presented an algorithm for establishing a correspondence between features of two faces. Facial features are obtained by a segmentation of the range image based on the sign of the mean and Gaussian curvature at each point. Gordon [64] identifies the nose region, ridge and valley lines by using mean and Gaussian curvature. Tanaka *et al.* [65] also perform curvature-based segmentation and represent the face surface using an Extended Gaussian Image (EGI). Basically, these approaches use the invariant functions, *e.g.* Gaussian curvature is invariant under Euclidean transformations, to extract information from the face surface and then perform classification based on the extracted information. The invariant functions they use all belong to the category called *differential invariants* which rely on derivative operations. The computation of differential invariants is not reliable because of the quantization error in discretized data. This fundamental issue limits their potential to achieve high recognition performance in a large dataset.

In [66], Lee *et al.* propose a novel implementation of a person verification system based on depth-weighted Hausdorff distance (DWHD) using the surface curvatures of the human face. Their method incorporates the depth information and curvatures of facial surfaces. The maximum, minimum and Gaussian curvatures can achieve recognition rates of 95%, 98% and 95%, respectively, for ranked best 5.

In [67], Lee *et al.* present a 3D face recognition system based on geometrically localized facial features. They propose the feature extraction procedure using the geometrical characteristics of a face. These features are directly applied to two face recognition algorithms, depth-based DP (Dynamic Programming) and feature-based SVM (Support Vector Machine). Experimental results show that recognition rates based on the depth-based DP and the feature-based SVM are 95% for 20 people and 96% for 100 people, respectively.

In [68], Moreno *et al.* analyze the discrimination ability of 3D descriptors extracted from facial surfaces. Segmentation of facial surface is done by using the signs of mean and Gaussian curvatures. Eighty six descriptors have been obtained from the segmented regions. Thirty five more discriminating ones in frontal views provide 78% of recognition rate.

More recently, Medioni *et al.* [69] perform 3D face recognition using iterative closest point (ICP) matching. Hesher *et al.* [70] perform principal component analysis (PCA) of range images. Similarly, Chang *et al.* [71] use a PCA-based method separately on 2D pictures and 3D range images. Matching scores from 2D and 3D are then combined for multi-modal recognition. Further investigation into the 3D-PCA approach has been reported by Heseltine *et al.* [72]. Kakadiaris *et al.* [73] present a 3D face recognition approach that fits a deformable model to the probe face surfaces. They perform experiments on the FRGC v1.0 dataset and report a 97% verification rate measured at a 0.1% false accept rate (FAR). Note that none of these recent results utilizes any invariant feature from the face surface.

Expression is a relatively unaddressed issue in face recognition literature. Chang *et al.* [74] tackle the issue of expression changes by utilizing three different regions around a nose, as being relatively rigid areas across different facial expressions. Passalis *et al.* [75] propose a fully automatic 3D face recognition algorithm based on the elastically adapted deformable model framework. Their algorithm can fit an Annotated Face Model (AFM) to an input 3D facial surface in the presence of facial expressions. A 3D face recognition approach based on geometric invariants was introduced by Bronstein *et al.* [76, 77]. The key idea of their algorithm is to approximate facial expressions as isometric transformations, *i.e.* length preserving transformations. Their experiments are performed on a dataset containing 220 faces of 30 subjects and 1.9% equal error rate is reported. Their system can distinguish between two identical twins based on 3D images of their faces.

Table 4.2 3D Face Recognition Algorithms Using Invariant Features

Author, year, reference	number of subjects	number of images	resolution of image	invariant feature	reported performance
Cartoux 1989 [62]	5	18	N.A.	principal curvatures	100%
Lee 1990 [63]	6	6	256 × 150	mean & Gaussian curvatures	N.A.
Gordon 1992 [64]	26 training 8 testing	26 training 24 testing	N.A.	curvature	100%
Tanaka 1998 [65]	37	37	256 × 256	curvature	100%
Moreno 2003 [68]	60	420	2.200 points	mean & Gaussian curvature	78%
Lee 2004 [66]	42	84	240 × 320	curvature	98%
Lee 2005 [67]	100	200	varying	curvature	96%
Bronstein 2005 [77]	30	220	N.A.	Geodesic distance	100%

4.2 Face Databases

The appearance of a face is subject to a large number of variations, such as pose, illumination, expression, occlusion, time lapse. The development of a face recognition algorithm requires a sufficiently large database with controlled variations of these parameters. Also, in order to evaluate different face recognition algorithms, it is necessary to have a common database. Evidently, the availability of public face databases is important for the advancement of this field.

Along with the development of face recognition technology, many face databases have been collected. In this section, a comparison of some publicly available face databases is presented. The list of databases, shown below, is limited to those relatively large and commonly used by the research community.

The availability of evaluation methods has equal importance as that of the face database. The face databases facilitate the development of face recognition algorithms, and evaluation protocols allow performance differences between algorithms to be quantified. Basically, an evaluation protocol consists of two parts. The first is the rules for conducting an evaluation. For example, training data, testing data and experimental configurations should be specified. The second part is the format of the results that allows scoring, *i.e.* how the system performance should be computed. Until the FERET evaluation, there was no evaluation protocol that included a large dataset and a standard evaluation method. Other databases associated with an evaluation method are BANCA, FRGC and XM2VTS.

All of these databases provide recordings of regular 2D/texture images. Only a few, like MPI, FRGC 3D and XM2VTSDB, collect 3D information of facial surfaces. The MPI database records 3D data with a Cyberware laser scanner [78]. For 7 out of 200 subjects, their 3D data are available for download. In Extended M2VTS Database, a 3D model of each subject's head was acquired using a high precision stereo-based 3D camera. 3D models of 293 subjects are available. In FRGC 3D dataset, 3D images were acquired by a Minolta Vivid 900/910 series sensor. The Minolta Vivid 900/910 series is a structure light sensor that takes a range image and a registered color image.

Table 4.3 summarizes the key features of these databases.

- AR database [79]
- BANCA database [80]
- CMU Pose, Illumination, and Expression (PIE) database [81]
- Face Recognition Grand Challenge (FRGC) dataset [82]
- Face Recognition Technology (FERET) database [83]
- Face Database of the Max Planck Institute (MPI) for Biological Cybernetics [84]
- NIST Mugshot Identification Database (MID) [85]
- Extended M2VTS Database (XM2VTSDB) multi-modal face database [86]

Table 4.3 Comparison of 2D Face Databases

Database	number of subjects	number of total images	Image resolution
AR	116	3288	768 × 576
BANCA	208	N.A.	720 × 576
CMU PIE	68	41,368	640 × 486
FERET	1,199	14,051	256 × 384
MPI	200	14,051	256 × 256
NIST MID	1,573	3248	varying
XM2VTSDB	295	N.A.	720 × 576
FRGC 2D	466	28,804	1704× 2272 or 1200 × 1600
FRGC 3D	466	4,950	640 × 480

N.A. denotes the information is not available.

4.3 3D Face Recognition Using Summation Invariants

In this section, we exploit the feasibility of applying summation invariant features for 3D facial range image classification. Several key design issues are addressed: (a) To identify, among many possible variations of summation invariant features in the family, those features that yield highest performance. (b) To investigate the impacts of various processing methods, including scaling and re-sampling on the range data. (c) To explore proper dimensionality reduction methods that will enhance the computation efficiency. We have conducted extensive experiments using the Face Recognition Grand Challenge v1.0 dataset and the BEE (Biometric Experiment Environment) package. The proposed features yield significant performance improvement over the FRGC 3D baseline.

The rest of this section is organized as follows. Section 4.3.1 briefly introduces the 3D face database and the evaluation protocol. Section 4.3.2 illustrates our algorithm in detail. In Section

4.3.3, we present the experimental results and compare them with those of the FRGC 3D baseline algorithm.

4.3.1 3D Face Dataset and BEE

We use the Face Recognition Grand Challenge (FRGC) v1.0 dataset [82] to conduct face recognition experiments. FRGC is sponsored by the US National Institute of Standard and Technology (NIST) and other government agencies. Its testing data contains comprehensive 3D images of human faces. It also provides an XML-based evaluation protocol, called the Biometric Experimentation Environment (BEE), to document and describe several face recognition experiments. All our experiments are conducted using BEE.

The 3D data provided by FRGC v1.0 contains 275 subjects (1 to 8 range scans per subject) and a total of 943 range scans. Each range scan has a resolution of 640×480 pixels. A 3D image consists of a 2D texture image and a 3D shape image. In this work, we use only the 3D shape images in all experiments.

4.3.2 Single Region Face Recognition Algorithm

FRGC defines four experiments, among which only experiment 3 concerns the 3D face recognition task. Our experimental procedures follow closely what has been defined for the baseline algorithm provided by FRGC except for the following modifications. Details of experiment 3 are omitted here due to space limitation.

1. **Use 3D data only**: In order to focus only on 3D face recognition performance, the fusion of matching scores in experiment 3 is disabled so that 2D texture results are not included.

2. **Arc-length re-sampling**: For 2D curve invariants $\eta_{i,j}$, range data are re-sampled uniformly with respect to arc-length. Specifically, for each row on the range data, we first compute its arc-length and re-sample it uniformly with respect to arc-length. Then, we perform the same

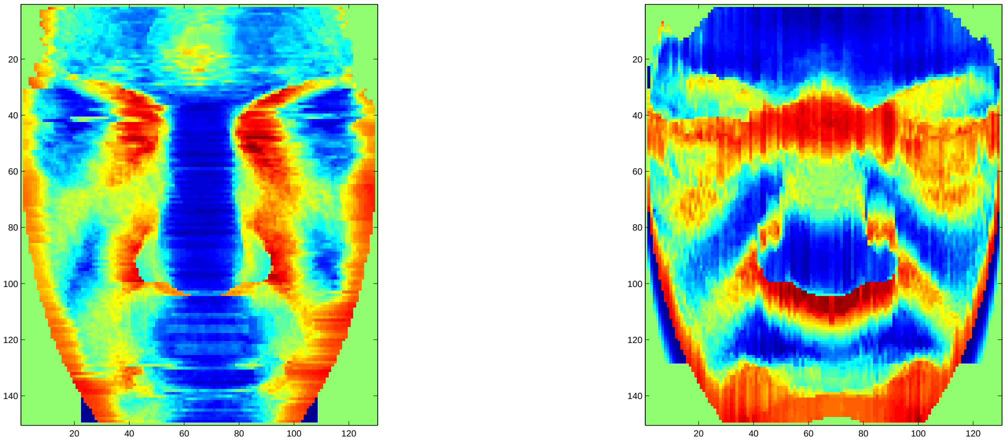
re-sampling on each column. For surface invariants $\kappa_{i,j}$, we do not perform re-sampling on the normalized range data.

3. **Semi-local summation invariants**: While the baseline algorithm uses 3D range data directly, we extract semi-local summation invariants from a normalized range image and use the results as invariant features. At each pixel, a semi-local summation invariant is computed from a local window. We compute curve invariants $\eta_{i,j}$ from both a horizontal window and a vertical window. The results are shown in Figure 4.1. The length of the local window is chosen to be $L = 21$. Similarly, we also compute the $\kappa_{i,j,k}$ from a local window surrounding each pixel. The window size for surface invariants is 17×17.

4. **Specify a region of interest**: Instead of using the entire 3D range image as the baseline algorithm does in BEE, we crop invariant features from an 81×81 rectangular region centered at the nose tip. A normalized range image with corresponding 81×81 cropped region is shown in Figure 4.2.

5. **Alignment refinement**: In FRGC, the location of the nose tip is manually selected. We use the provided nose tip location to define an initial 81×81 region. The summation invariants computed from the 81×81 region is called the *summation image*. For all the normalized range data, we compute their summation images from the initial region and find their average image, called *the mean of summation images*. Then, for each normalized range data, we find a new 81×81 region which has minimal SSD (sum of squared differences) with the mean of summation images and compute a summation invariant from the new region. This procedure leads to more accurate alignment of range data.

6. **Dimension reduction**: In order to reduce the size of the feature vectors, we use principal component analysis (PCA) to compute their subspace projections. The PCA basis is computed using the training set containing 183 range images. For classification purpose, the similarity metric is the Mahalanobis cosine. Note that in our experiments, the training set



(a)

(b)

Figure 4.1 (a) The computed result of $\eta_{0,1}$ using a horizontal window; and (b) The computed result of $\eta_{0,1}$ using a vertical window.

and the similarity metric are the same as those in the baseline algorithm. Please refer to [71] for more details

7. **Score fusion**: There are many ways of combining multiple classifiers to achieve a better recognition performance. Here, we use the sum rule to combine the matching scores produced by different invariant features.

The code is implemented in non-optimized C language. The experiments are conducted under a Linux operating system with a 3.40 GHz XEON processor and 2GB memory. Given the normalized range images, it takes about 5 minutes to compute $\eta_{i,j}$ and accomplish the following analysis (*e.g.* computing similarity metrics and performing ROC analysis). For computing $\kappa_{i,j,k}$ and the following analysis, it takes 8 minutes. For the currently available datasets (*e.g.* FRGC v2.0 dataset has $50,000$ images), computation cost will be one of the major concerns in designing a face recognition algorithm. We think summation invariant is an efficient and feasible solution to operate on the large scale datasets like FRGC v2.0.

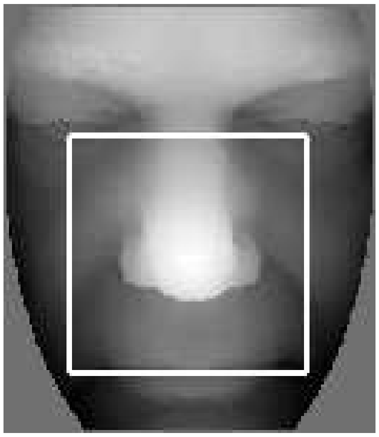


Figure 4.2 Normalized depth map and an 81×81 region centered at the nose tip.

4.3.3 Experimental Results

We have conducted a series of experiments to assess the performance of the proposed algorithm. The first part is to examine how the recognition performance is affected by the undesirable scaling in 3D data. The second part is to evaluate the discriminative capability of individual summation invariants. Decision fusion is considered in the third part. We use a simple fusion strategy to combine results at the metric level. Finally, we compare with other 3D face recognition algorithms which also operate on the FRGC v1.0 dataset.

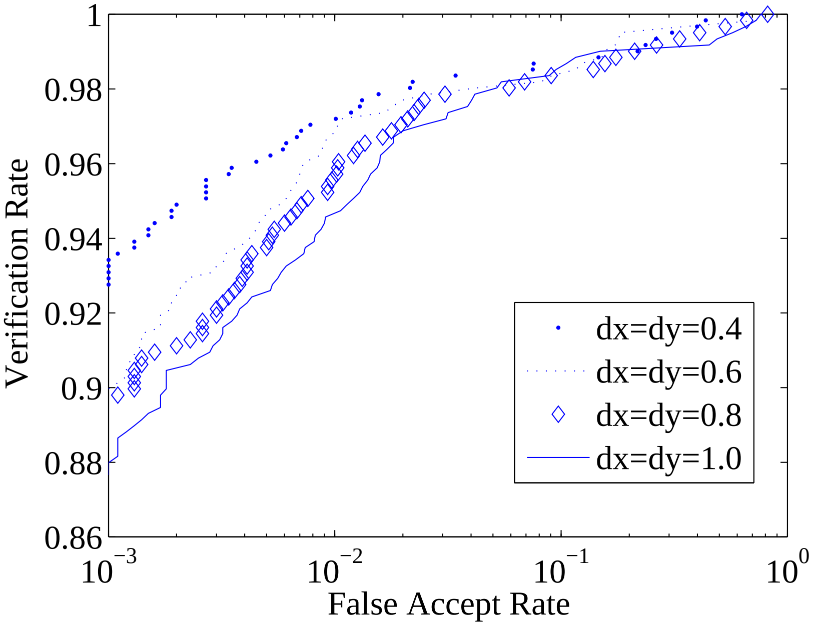
4.3.3.1 Effects of Difference Scaling of x and y Coordinates

In this experiment, we explore the issue of scaling on range data by using an individual summation invariant without decision fusion. Recall that the summation invariants developed in section 2.3 are for the Euclidean group acting on $\mathbb{R}^2$ or $\mathbb{R}^3$. While this feature is invariant to Euclidean transformations, it will be affected by improper scaling of coordinates. In the baseline algorithm, x and y coordinates in the range data are discarded after image normalization. Also, the z coordinates are multiplied by a scaling factor to enhance contrast of range images. In our method, however, it is very important to retain the original geometry of a facial surface. To assess the issue of improper scaling on range data, we experimented with different values of increments,

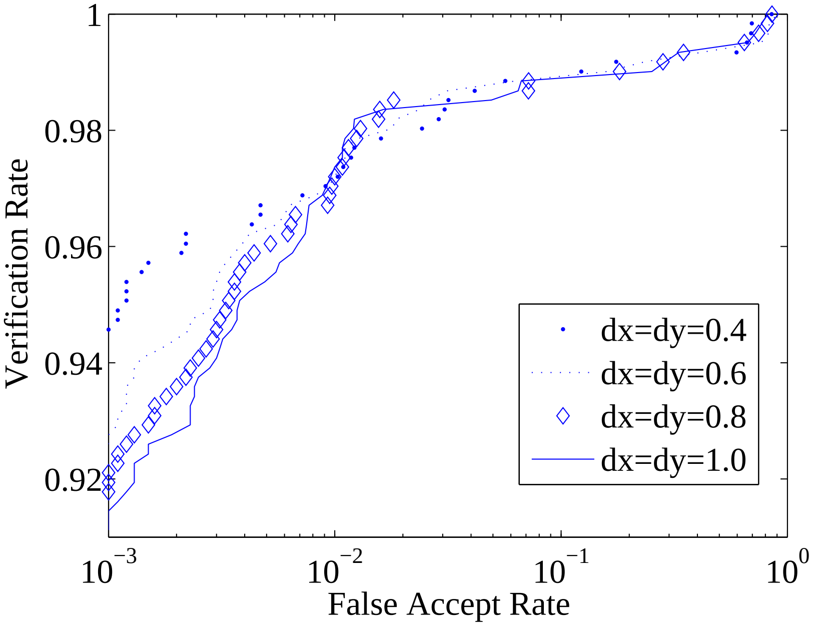
$dx = dy \in \{0.4, 0.6, 0.8, 1.0\}$, and a fixed scaling factor 12.5 on z values. In Figure 4.3, we observe a significant impact of these changes on the ROC curve. Specifically, by using either $\eta_{0,1}$ or $\kappa_{0,0,1}$, the verification rates increase when dx and dy decrease from 1.0 to 0.4. Then, the verification rates decrease when dx and dy are smaller than 0.4. Hence, we assume that $dx = dy = 0.4$ and a fixed scaling factor 12.5 on z values can reasonably restore the original geometry of a face surface and use these parameters in the following experiments. In general, better recognition performance is expected if the original x, y, z information is preserved.

4.3.3.2 Effects of Different Summation Invariants

Summation invariants are functions which are unaffected under the actions of a particular transformation group. In other words, they do not guarantee any discriminative power for object recognition purpose. Hence, it is important to know how they perform individually in a face recognition application. In this experiment, we do not use score fusion so that the contribution of an individual summation invariant can be identified. From the ROC curves in Fig 4.4, it is apparent that not all semi-local summation invariants are created equal in terms of discriminating power. Among five 2D summation invariants, $\eta_{1,1}$ yields uniformly highest verification rate at different false accept rate. Among five 3D summation invariants, $\kappa_{0,0,1}$ produces uniformly highest performance. By the way, summation invariant $\eta_{1,1}$ achieves the same level of recognition performance as those of summation invariants $\kappa_{0,0,1}$ while their computation costs are quite different. The computational cost of $\eta_{i,j}$ and $\kappa_{i,j,k}$ are $O(L)$ and $O(L^2)$, respectively, where L is the size of the local window. The results also raise an interesting question about why some invariant features are better in term of recognition performance. Are there some approaches to predict their performance in advance? An answer to this question will lead us to construct a better invariant feature. On the other hand, we should be able to further improve recognition performance by simply combing these invariant features.

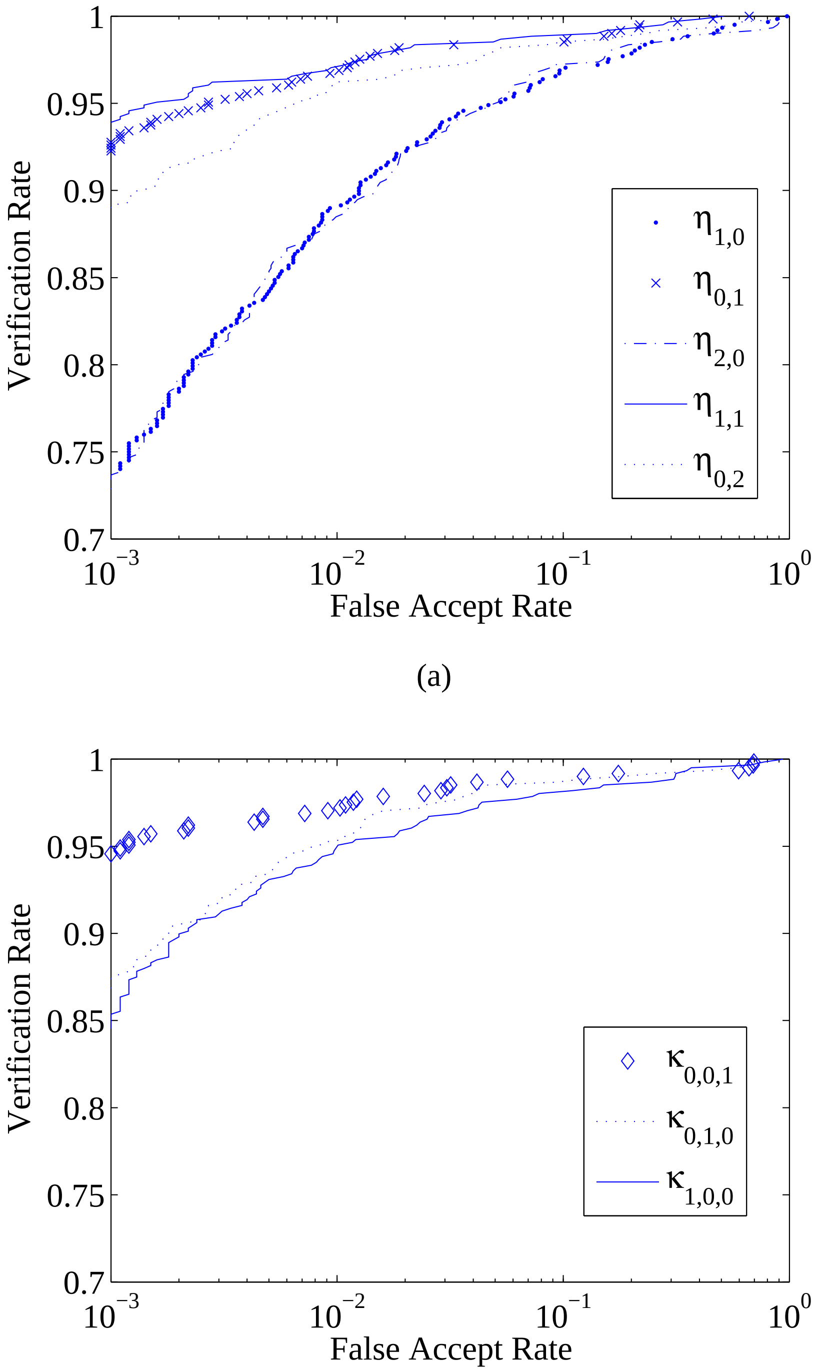


(a)



(b)

Figure 4.3 ROC performance for using different unit difference on the x and y coordinates. The results indicate that geometrical distortion will cause performance degradation of summation invariants. (a) ROC curves of $\eta_{0,1}$ and (b) ROC curves of $\kappa_{0,0,1}$.



(a)

(b)

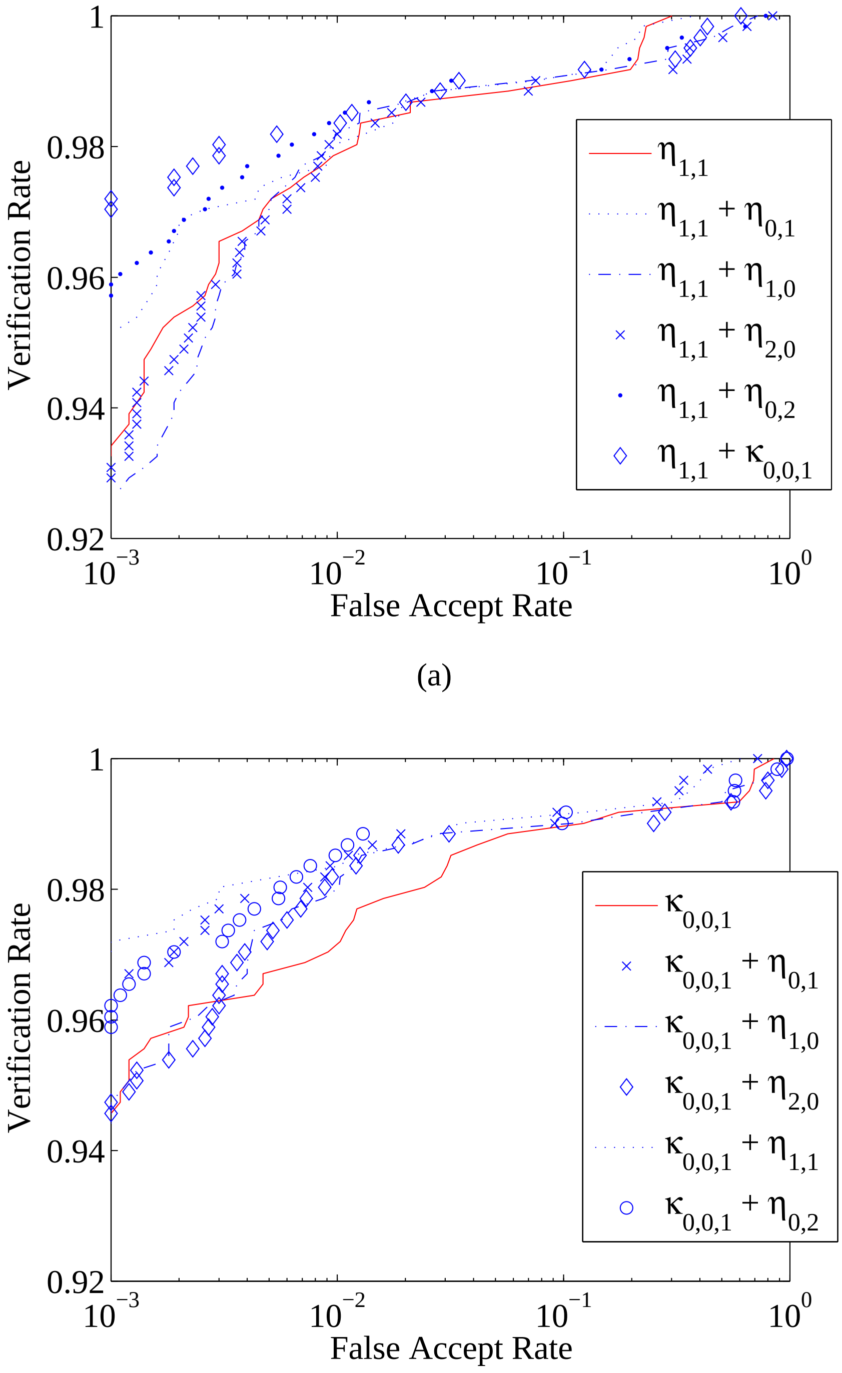
Figure 4.4 ROC performance obtained using different summation invariants. (a) $\eta_{i,j}$ and (b) $\kappa_{i,j,k}$

4.3.3.3 Effects of Combining Summation Invariants

This experiment investigates the value of using more than one summation invariant. The hypothesis for this experiment is that the combined decision of these summation invariants will further improve the face recognition accuracy. We only choose summation invariants which have higher recognition rates to perform fusion because there are too many possible combinations to be shown. Figure 4.5 shows the ROC curves of combining two summation invariants. The fusion strategy is simply adding the similarity scores from two different summation invariants, *i. e.* equal weights. We observe that fusion does not always yield a higher verification rate. For example, the combination of $\eta_{1,1}$ and $\eta_{1,0}$ results in performance degradation at low false alarm rate region. Kung *et al.* [87] introduce the notation of **consistent fusion**, which by definition has an equal or better performance than any individual classifier in the entire FAR region. Here, the *inconsistent fusion* results are due to the improper weightings on classifiers. In the next chapter, we will focus on the issue of information fusion and present other fusion strategies with consistently superior performance to that of direct fusion.

4.3.3.4 Comparison With Other 3D Face Recognition Algorithms

In this section, we conduct three experiments to evaluate the performance of our algorithm and the FRGC baseline algorithm. In the first, we simply run the FRGC baseline algorithm on 3D data alone. In the second, we still run the FRGC baseline algorithm on 3D data but using only the 81×81 cropped region rather than the whole normalized range data. In Fig 4.6, the second experiment yields a lower verification rate than the first one. This is reasonable because the second experiment uses less data to perform recognition. In the third experiment, we compute $\eta_{1,1}$ and $\kappa_{0,0,1}$ from the cropped region and then combine them by the sum rule. The proposed algorithm yields the highest verification rate in the entire FAR region as one can see in Fig 4.6. It achieves the verification rate of 97.2% at the false accept rate of 0.1% and the EER (Equal Error Rate) of 1.43%. The results clearly indicate that summation invariants offer statistically significant better



(a)

(b)

Figure 4.5 ROC performance obtained by combining two summation invariants. (a) Fusion of $\eta_{1,1}$ and others. (b) fusion of $\kappa_{0,0,1}$ and others.

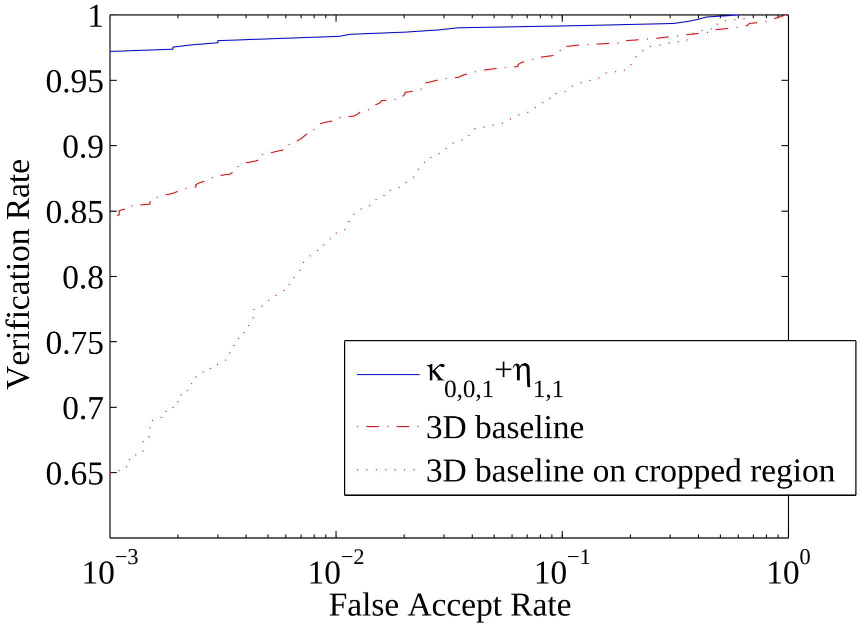


Figure 4.6 Comparison with FRGC 3D baseline algorithm. We apply the FRGC 3D baseline algorithm on the normalized depth map and the cropped region shown in Fig 4.2. Their corresponding ROC curves are shown by the solid line and the dash line respectively.

recognition performance than the range data itself. Note that these three experiments use exactly the same PCA parameters, specifically dropping the first 10 eigenvectors and using the following 100 eigenvectors.

Except for the FRGC 3D baseline, some recent results on the FRGC v1.0 dataset are briefly discussed below. Kakadiaris *et al.* [73] present an approach based on the Annotated Face Model (AFM) and report about 97% verification rate at 0.1% false accept rate. Pan *et al.* [88] propose a novel mapping of the range image, called *mapped relative depth image*, and use it as a feature to perform classification. Their method achieves an EER of 2.83% on the FRGC v1.0 dataset. Russ *et al.* [89] develop a 3D face recognition approach based on the Hausdorff distance metric. Their experimental validations are conducted on part of the FRGC v1.0 dataset, using only a single probe per person rather than all available probes. Performance is reported as a verification rate of 93.5%

at a false accept rate of 0.1%. Evidently, the proposed method has comparable performance with the state-of-the-art algorithms for 3D face recognition.

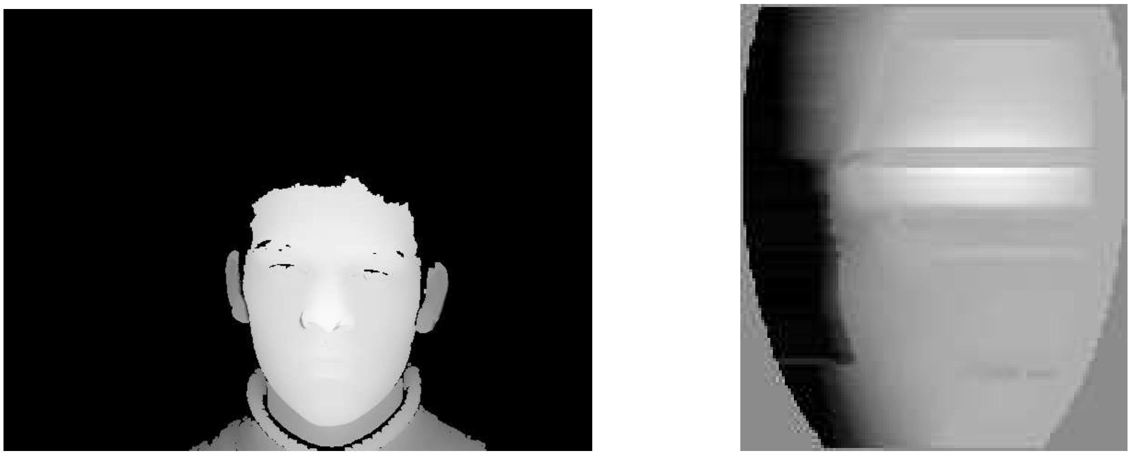
4.4 Summary

The value of summation invariants in the context of 3D face recognition is evaluated in this chapter. The geometric features of a facial surfaces are extracted by using summation invariants and then the dimensionality is reduced by PCA on the resulting representation. To the best of our knowledge, Kakadiaris *et al.* [73] report the best previous results using the FRGC v1.0 dataset. The proposed algorithm can yield the same level of recognition performance. Furthermore, such a good performance can be achieved using only the nose portion of the whole face. In general, the results support the following conclusions:

1. Geometric features extracted by using summation invariants provide useful information for recognition purposes.

2. The combination of two or more summation invariants generally improves performance over using a single summation invariant.

Also, we should note that the results reported in this work are obtained using the normalized range images provided by the FRGC 3D baseline. In theory, we can compute summation invariants from range images without normalizing position and orientation while maintaining the same performance. Recall that summation invariants are designed to be unaffected by rotations and translations at the very beginning. Currently, the normalization of raw images is the most time-consuming part in the FRGC 3D baseline. The normalization alone takes about 20 minutes and the rest takes about 5 minutes. Thus, an important part of future research along this direction is the development of a more efficient 3D face recognition technology in which normalization efforts are reduced by using summation invariants.



(a) (b)

Figure 4.7 Failure example of the FRGC 3D normalization process: (a) Original range image and (b) Normalized range image.

Some recognition errors reported by our algorithm are due to the failure in the normalization step (Figure 4.7). The baseline normalization algorithm works for most of the range images but not in some special cases. Note that the general quality of 3D range images is not as good as 2D pictures. In some regions, like eyes, cheeks and eyebrows, current 3D sensing devices, such as the Minolta vivid 900, tend to either produce wrong depth values or have missing data. These issues would limit the 3D verification rate in general. Algorithms for reliably recovering data in such areas is an area where more research is needed.

Chapter 5

Optimal Linear Fusion for Improving Performance

In this chapter, we propose a novel multi-region face recognition algorithm. Specifically, we identify multiple sub-regions over a given facial image, and extract summation invariant features from each sub-region. For matching an image pair, the summation invariant feature of each sub-region produces a matching score. Then, linear optimal fusion schemes are used to integrate matching scores from each sub-region. Earlier, in chapter 4, we introduced a single-region face recognition algorithm. Here, we will focus on the development of a multi-region 3D face recognition method that incorporates novel feature fusion and feature selection methods to significantly enhance the performance.

The 3D face recognition experiment is conducted using *Face Recognition Grand Challenge* (FRGC) version 2.0 data set under the BEE (Biometric Experimental Environment) protocol. During the training stage, a 943 by 943 similarity matrix is computed from the training partition of the FRGC 3D dataset. The $(i, j)^{th}$ entry of this similarity matrix represents a measure of how close the feature vector extracted from the i^{th} gallery image is to the j^{th} probe image. It is often a real number between 0 and 1 after score normalization. With N sub-regions and hence N different features extracted from the same face image, the objective of score fusion is to combine the N corresponding similarity scores into a final similarity score. Toward this goal, we explore two linearly optimal fusion methods, namely *Linear Discriminant Analysis* (LDA) and *Linear Support Vector Machine* (LSVM).

The rest of this chapter is organized as follows: In section 5.1, we give a brief review of previous works on information fusion. Section 5.2 presents the experimental setup and results of three fusion schemes, namely *sum rule*, *Linear discriminant Analysis* and *Linear Support Vector Machine*. Finally, we make some concluding remarks in section 5.3.

5.1 Literature Review

In the pattern classification literature, it is well known that combining multiple classifiers often leads to superior performance to that achieved by any individual classifier [90–92]. There are two major categories of classifier fusion architecture: **stacked generalization** [93], also known as the *committee machine* [94–97], or the *ensemble method* [98–103]; versus **mixture of experts** [104–106].

Previously, in [107], optimal data fusion is presented under the constraint of a fixed k out of n weighted threshold fusion architecture. In [108], a hierarchical model is used and the Bayesian Gibbs sampling method is used to design the fusion rule. Data fusion has also been studied in the context of combining multiple classifiers. In [109], three types of classifier combination methods, namely, *averaged Bayes classifiers*, *voting principals*, and *Dempster-Shafer fuzzy combinations* have been reviewed. Some experiments have been conducted but no conclusive comparison results are available. In [110], the accuracy of individual classifiers are estimated, and classifiers are selected dynamically based on which classifier will yield the best performance in a specific local region. Ji and Ma [111] proposed to use a structure consisting of randomly generated linear local classifiers with a voting fusion mechanism to perform pattern classification tasks. Petrakos *et al.* [112] discussed the effect of correlations between classifiers and their impacts on fusion performance. The Behavior Knowledge Space (BKS) method [113] is a non-parametric decision fusion method that uses a look-up-table (LUT) to implement the decision fusion classifier. For the given

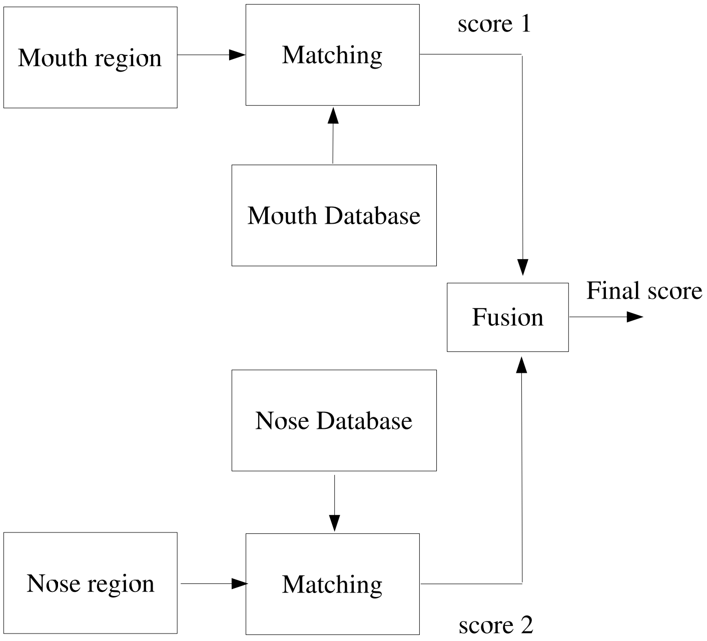


Figure 5.1 Example of opinion/score fusion. The scores from mouth region and nose are combined to yield a final score.

set of training samples, the BKS method is guaranteed to offer the best decision fusion results. However, with a finite number of training samples, the BKS method does not necessarily yield the best generalization results.

Information obtained from multiple resources could be fused using a number of approaches, at different levels. We can create a more reliable system by integrating information from multiple resources. In general, there are four possible levels of fusion [114]: (1) sensor level, (2) feature level, (3) opinion/score level, (4) decision level. Figure 5.1 shows an example of fusion at opinion/score level.

There might be a strong correlation between the information resources (*e.g.*, multiple invariant features on a face image) and such a scenario is referred to as tightly coupled integration [115]. On the other hand, the loosely coupled system means little correlation between information resources (*e.g.*, feature representation from mouth and nose) and fusion occurs at the output of independent classifiers. It has been shown that the independence of classifiers plays an important role in performance improvement [116]. Hence, in the following sections, we will focus on the analysis of the loosely coupled system.

5.2 Multi-Regions 3D Face Recognition Algorithm

As one can see in the previous chapter, the single-region algorithm works very well on the FRGC v1.0 dataset [117], which contains only neutral expression. Unfortunately, we observe a significant performance drop when it is applied to FRGC v2.0 dataset where expression changes exist. To enhance the performance of the single-region algorithm, the major difficulty is to find a single region which is rigid across expressions and also large enough to provide good discriminating power. A natural way to overcome this limitation is to utilize multiple regions. In a multi-region algorithm, each region should be relatively small compared to the one used in the single-region algorithm so that it is relatively rigid across expressions. Also, a multi-region algorithm is expected to perform better than combining multiple summation invariants [117]. If we compute different summation invariants from a single region, the extracted information will be highly correlated. Hence, the benefits of integrating multiple summation invariants will be limited. On the other hand, the information from different facial regions will be more likely uncorrelated. We will only address the fusion of different facial regions, referred to as the *loosely coupled system* [115], rather than the fusion of different summation invariants.

The scenario of fusion is as follows: Each single-region sub-system matches an input image pair independently and generates the corresponding matching score. The matching scores of multiple single-region sub-systems are combined by different fusion schemes. Each sub-system is based on one of the 10 regions shown in Fig. 5.2. We will discuss three approaches for information fusion.

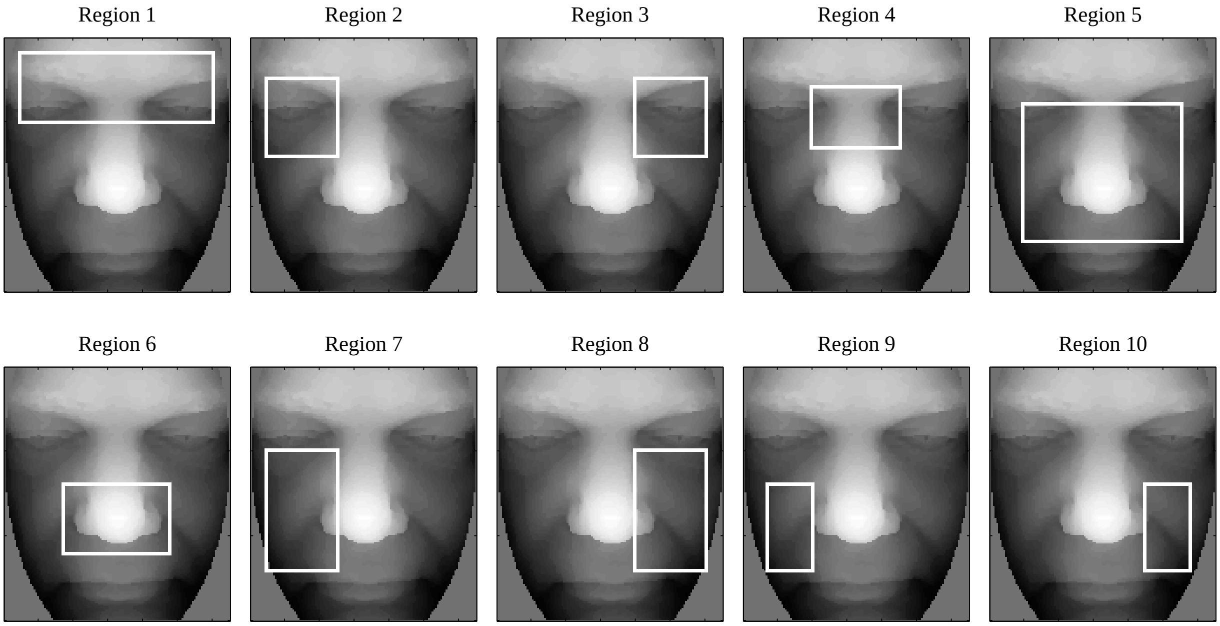


Figure 5.2 We specify 10 regions on facial surface. Matching scores obtained from each region are combined to yield the final matching score.

5.2.1 Description of Experiments

We perform experiments using the FRGC v2.0 dataset [82]. In the FRGC, 3D images with a resolution of 640×480 consist of both shape and textures channels. The disjoint training and validation partitions contain 943 and 4007 3D images, respectively. Among the challenge problems defined in FRGC v2.0, we focus on experiment 3s, which utilizes only the shape channel in a 3D image. Performance will be reported on a receiver operator characteristic (ROC) curve that shows the trade-off between verification and false accept rate. In the FRGC protocol, three masks are defined over the similarity matrix where each entry contains the matching score of an image pair. Each mask collects its own set of entries in the similarity matrix, thus generating three ROC curves which will be referred to as ROC I, II and III. The images within an image pair are called gallery and probe. In ROC I, gallery and probe are from the same semester. In ROC II, gallery and probe are from the same year. In ROC III, gallery and probe are from different semesters. On average, ROC III has the longest time lapse between gallery and probe images and therefore is the most challenging.

5.2.2 Sum Rule

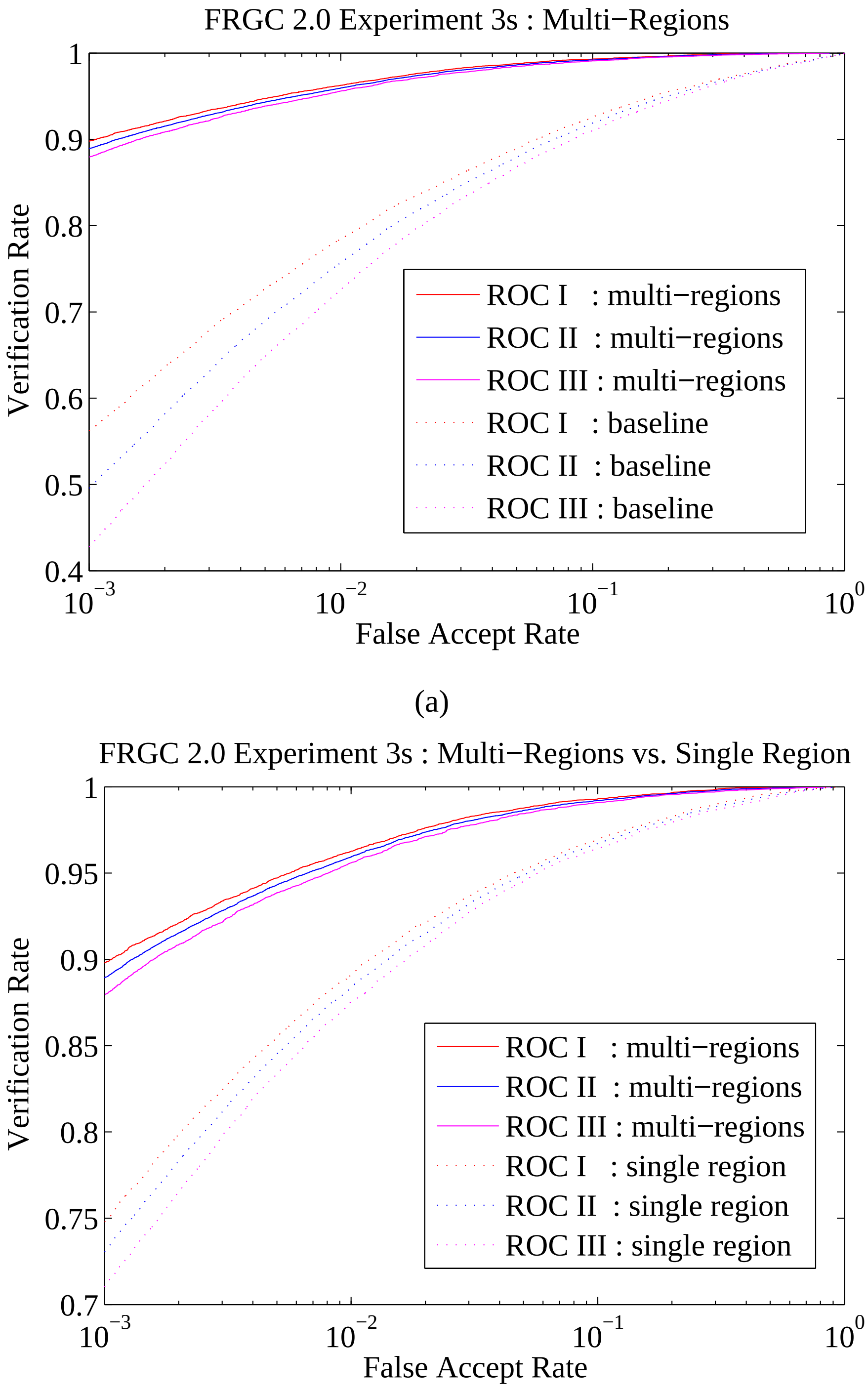
The simplest method of fusion is to add the scores from multiple regions. Equal weights are assigned to each region since we don't know the relative importance of each region. Figure 5.3 shows the results of using the sum rule on the FRGC v2.0 dataset. Note that the verification performance of the single-region algorithm is better than the BEE baseline. This, once again, confirms the fact that summation invariants provide better discriminating capability than depth value does. But, the performance improvement is limited in the presence of facial expressions. This limitation can be overcome by utilizing multiple regions. By combining relatively small regions which are relatively unaffected by expressions, we observe significant improvement in the verification rate.

5.2.3 Linear Discriminant Analysis

In the FRGC protocol, there are two non-overlapping data partitions, training and validation. During algorithm development, experiments are conducted on the data in the validation partition. This allows researchers to tune the parameters of their approaches. However, finding a subspace representation and classifier training are required to be conducted on the training partition. For example, PCA-based algorithms should construct eigen-faces from the training set. Here, we use Linear Discriminant Analysis (LDA) [118] to obtain an optimal weighting vector from the training partition. In the FRGC v2.0 dataset, there are a total of 943 3D images in the training partition.

Let $S = (s_1, s_2, \ldots, s_n)$ denote a score vector, where n is the number of facial regions ($n \leq 10$ in our experiments). Each s_i is the matching score of i-th region, generated from a pair of images. The LDA is a projection from the original score vector $S \in \mathbb{R}^n$ to final matching score $\widetilde{S} \in \mathbb{R}$.

$$\widetilde{S} = W_{LDA}^T S \tag{5.1}$$



(a)

(b)

Figure 5.3 ROC performance obtained by using sum rule for information fusion. (a) Comparison with BEE baseline algorithm and (b) Comparison with single-region algorithm

where W_{LDA} is the $n \times 1$ weighting vector obtained by solving

$$W_{LDA} = \arg\max_{W} \frac{W^T S_B W}{W^T S_W W} \tag{5.2}$$

where S_B is the between-class scatter matrix and S_W is the within-class scatter matrix.

$$\begin{aligned} S_B &= \sum_{i=1}^{2} N_i (m_i - m)(m_i - m)^T & (5.3) \\ S_W &= \sum_{i=1}^{2} \sum_{S_k \in \mathbb{S}_i} (S_k - m_i)(S_k - m_i)^T & (5.4) \end{aligned}$$

In Equation (5.3) and (5.4), m is the mean score vector of all the training samples, m_i is the mean score vector of samples belonging to class i, $\mathbb{S}_i$ denotes the set of samples belonging to class i, and N_i is the number of training samples in class i. The solution for the criterion Equation (5.2) is given by

$$W_{LDA} = S_W^{-1}(m_1 - m_2) \tag{5.5}$$

In LDA-based face recognition such as [56], the number of classes is the number of subjects. Here, we apply LDA on a two class problem, namely match class and non-match class. Match class contains the score vectors from image pairs of the same person while non-match class is generated by image pairs of different persons. The LDA provides an optimal projection direction that maximizes the separation between two classes. Figure 5.4 shows the ROC curves of using LDA. We observe a significant improvement over the sum rule. The results suggest that the distribution of training data is important knowledge available for use in constructing a better classifier. We should also mention the fact that the training partition contains only neutral expressions while the validation partition contains different expressions. Hence, the distribution of training samples is expected to be different from the distribution of validation data in the FRGC v2.0 protocol. This

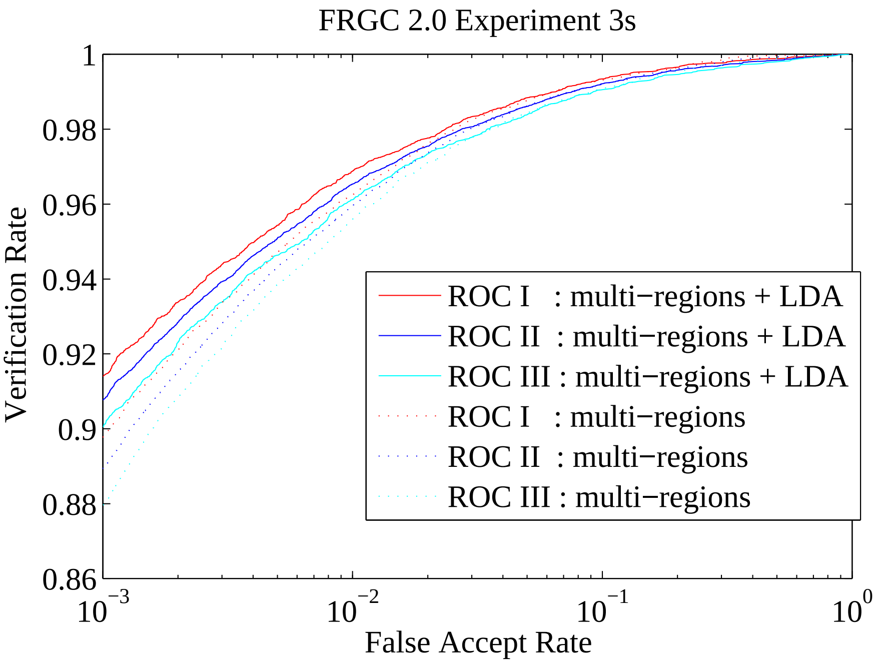


Figure 5.4 ROC performance obtained by using LDA for information fusion.

also points out a future direction where one could expect performance improvement by including expression changes in the training samples. The weights obtained by LDA are shown in Table 5.1. From the table, we observe that LDA puts the highest weight on the nose region and the second highest one on the forehead region. It indicates that the nose and forehead are the two most important features in 3D face recognition with variation in expression.

5.2.4 Linear Support Vector Machine

Linear support vector machine (LSVM) is the simplest kind of SVM. It is a linear classifier with maximum margin. The *margin* means the width that a decision boundary can be increased to without contacting a data point. By maximizing margin, it gives us the least chance of making a misclassification. Consider a set of d-dimensional training data belonging to two classes. The decision hyperplane and margins are given by

Table 5.1 The Weights on Selected Regions Obtained by LDA

Number of	Weights on selected regions									VR
regions	#1	#2	#3	#4	#5	#6	#7	#8	#9	ROC-III
9 regions	.358	.322	.266	.366	.484	.400	.239	.274	.183	.9045
8 regions	.363	.316	.267	.378	.495	.406	.256	.276	0	.8910
7 regions	.370	.312	.270	.375	.622	0	.288	.281	0	.8614
6 regions	.392	.340	0	.390	.628	0	.299	.304	0	.8918
5 regions	.398	.373	0	.392	.667	0	0	.319	0	.8983
4 regions	.405	.375	0	.395	.733	0	0	0	0	.8663
3 regions	.469	0	0	.437	.767	0	0	0	0	.8133
2 regions	.536	0	0	0	.843	0	0	0	0	.8414
1 region	1.00	0	0	0	0	0	0	0	0	.7242

Verification Rate (VR) is measured at the point where False Accept Rate equals 0.1%.

$$\text{Plus plane} = \{\mathbf{x} : \mathbf{W}_{LSVM}^T\mathbf{x} + b = 1\} \tag{5.6}$$

$$\text{Decision plane} = \{\mathbf{x} : \mathbf{W}_{LSVM}^T\mathbf{x} + b = 0\} \tag{5.7}$$

$$\text{Minus plane} = \{\mathbf{x} : \mathbf{W}_{LSVM}^T\mathbf{x} + b = -1\} \tag{5.8}$$

where $\mathbf{x}, \mathbf{W}_{LSVM} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. The margin width M is defined as the shortest Euclidean distance between plus and minus plane.

$$M = \frac{2}{\|\mathbf{W}_{LSVM}\|} \tag{5.9}$$

So, we just need to search for the $\mathbf{W}_{LSVM}$ and b with the widest margin. This can be formulated as an optimization problem as follows: given the training data with labels $\{x_k, y_k\}, k = \{1, \ldots, N\}, y_k \in \{-1, 1\}, x_k \in \mathbb{R}^d$,

$$\min \|\mathbf{W}_{LSVM}\| \quad \text{subject to} \quad y_k(\mathbf{W}_{LSVM}^T\mathbf{x} + b) \leq 1 \quad \forall k \tag{5.10}$$

Note that this formulation consider the cases where data is linearly separable. It can be extended to linear non-separable cases, which is assumed in our research by taking classification errors into account.

$$\min \|\mathbf{W}_{LSVM}\| + C\sum_{k=1}^{N} \epsilon_k \tag{5.11}$$

subject to

$$\begin{aligned} \mathbf{W}_{LSVM}^T\mathbf{x} + b &\geq 1 - \epsilon_k \quad \text{if } y_k = 1 &(5.12)\\ \mathbf{W}_{LSVM}^T\mathbf{x} + b &\leq -1 + \epsilon_k \quad \text{if } y_k = -1 &(5.13)\\ \epsilon_k &\geq 0, \quad \forall k &(5.14) \end{aligned}$$

where C is a penalty factor assigned by user and ϵ_k is classification error. For the data points that have been classified correctly, their ϵ_k is zero. The details about how to solve this optimization problem are beyond the scope of this paper. Readers may refer to [119] for more detailed discussion on solving optimization problems.

We use the software package LIBSVM [120] to perform support vector classification. Similar to LDA, the score vector S is projected along the direction $\mathbf{W}_{LSVM}$ to yield the final score.

$$\widetilde{S} = \mathbf{W}_{LSVM}^T S \tag{5.15}$$

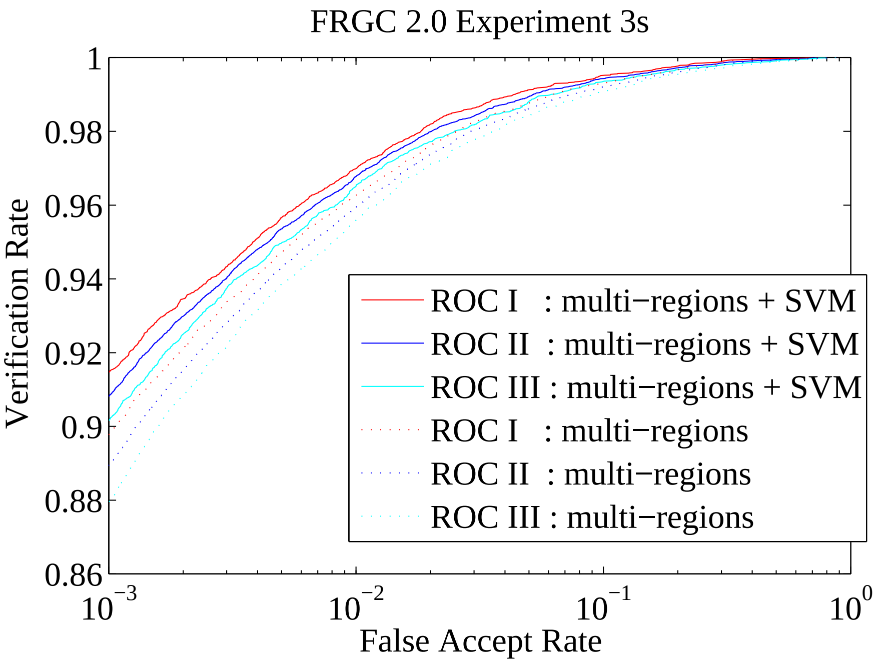


Figure 5.5 ROC performance obtained by using LSVM for information fusion.

Figure 5.5 shows the ROC curves of fusion by LSVM. The LSVM classification yields similar performance as LDA but is computationally more expensive. The weights obtained by LSVM are shown in Table 5.2. Similar to the LDA results, we observe that LSVM puts the highest weight on the nose region and the second highest one on the forehead region. It confirms the importance of the nose and forehead in 3D face recognition.

5.2.5 Nonlinear Fusion

The linear fusion schemes described in previous sections can only yield linear decision boundaries to separate the matching and non-matching classes. To allow more flexible decision boundaries, one can use nonlinear classifiers to combine the matching scores. For example, support vector machines with nonlinear kernels provide better decision boundaries to fit training data. Recall that the decision function for linear SVM has the following form:

Table 5.2 The Weights on Selected Regions Obtained by LSVM

Number of	Weights on selected regions										VR
regions	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	ROC-III
10 regions	2.53	1.29	1.42	1.99	2.79	1.88	1.44	1.50	.826	1.03	.9014
9 regions	2.51	1.29	1.49	1.81	2.80	2.10	1.56	1.66	0	1.20	.9014
8 regions	2.28	1.25	1.65	1.89	3.10	2.17	1.87	1.78	0	0	.9028
7 regions	2.45	0	1.78	2.01	3.07	2.29	1.99	1.74	0	0	.9012
6 regions	2.60	0	1.90	1.97	3.70	2.34	2.24	0	0	0	.9030
5 regions	3.04	0	0	2.32	3.96	2.51	2.25	0	0	0	.8978
4 regions	3.68	0	0	0	4.37	2.51	2.31	0	0	0	.8715
3 regions	3.96	0	0	0	5.24	2.82	0	0	0	0	.8656
2 regions	4.31	0	0	0	6.91	0	0	0	0	0	.8439
1 region	0	0	0	0	1	0	0	0	0	0	.7242

Verification Rate (VR) is measured at the point where False Accept Rate equals 0.1%.

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i \alpha_i \mathbf{x}^T \mathbf{x}_i + b \tag{5.16}$$

where **x** denotes a feature vector to be classified, $\mathbf{x}_i$ denote support vectors, y_i denotes the labels of support vectors, N is the total number of support vectors, α_i denote Lagrange multipliers and b is a bias term. The values of the Lagrange multipliers $\{\alpha_i\}$, support vectors $\{\mathbf{x}_i\}$ and bias b are obtained from training data.

To obtain a nonlinear decision boundary, usually resulting in better classification results, one can replace the inner product $\mathbf{x}^T\mathbf{x}_i$ with a nonlinear kernel $K(\mathbf{x}, \mathbf{x}_i)$.

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{5.17}$$

The basic idea behind the nonlinear SVM is to map the original feature vectors to a higher-dimensional feature space by using a kernel function. The data points might become linearly separable after mapping to a higher-dimensional feature space. There are various kinds of nonlinear kernels. The most common kernel functions are polynomial, Radial Basis Function (RBF) and Sigmoidal kernels.

$$\text{Polynomial Kernel} \quad : \quad K(\mathbf{x}, \mathbf{x}_i) = (1 + \frac{\mathbf{x}^T \mathbf{x}_i}{\sigma^2})^p, p > 0 \tag{5.18}$$

$$\text{RBF Kernel} \quad : \quad K(\mathbf{x}, \mathbf{x}_i) = exp\left\{ - \frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2} \right\} \tag{5.19}$$

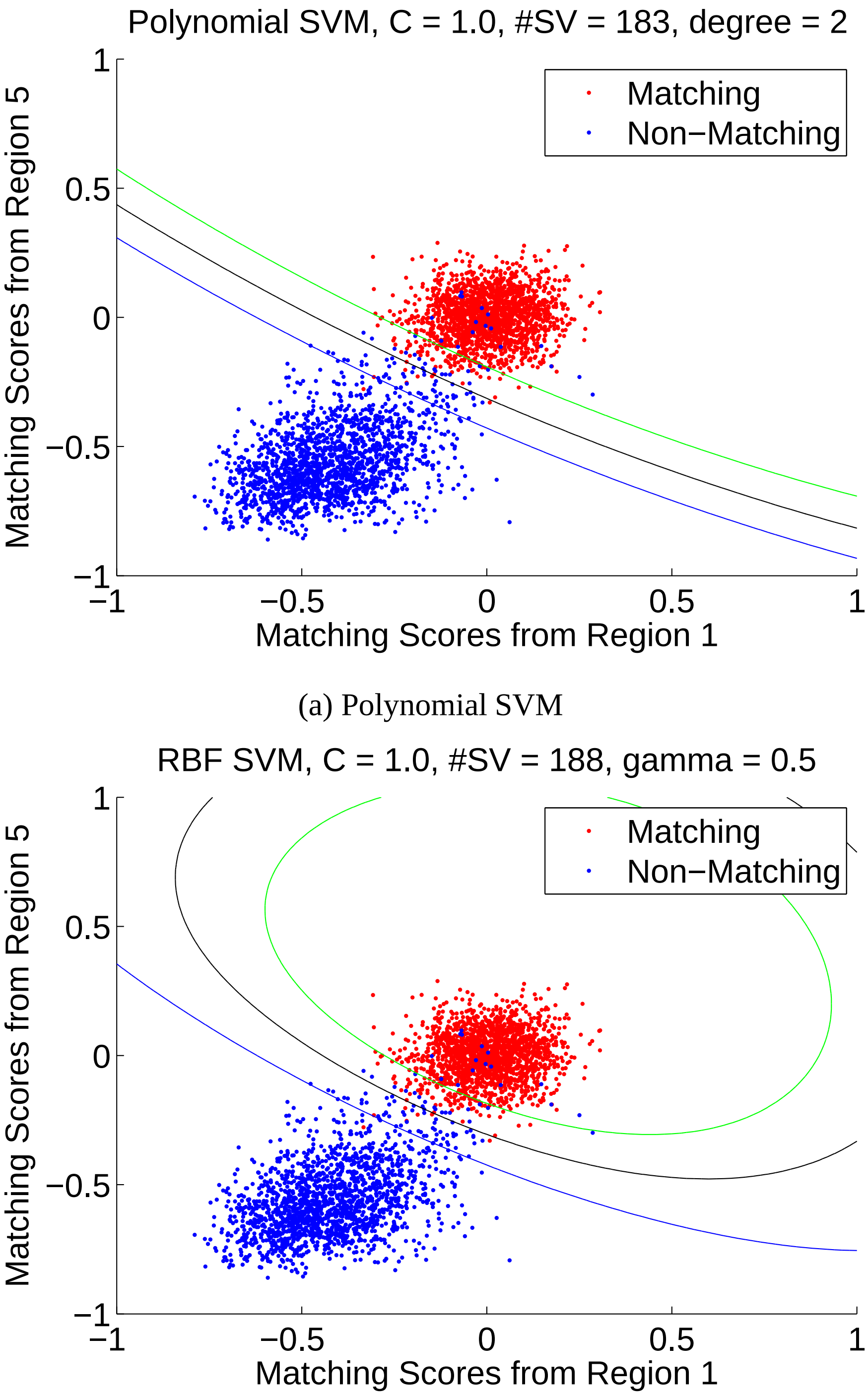
$$\text{sigmoidal Kernel} \quad : \quad K(\mathbf{x}, \mathbf{x}_i) = \frac{1}{1 + exp\{-\frac{\mathbf{x}^T \mathbf{x}_i + b}{\sigma^2}\}} \tag{5.20}$$

Again, we use the software package LIBSVM [120] to train SVMs with different kernel functions. The decision boundaries created by polynomial and RBF SVMs are shown in Figure 5.6. Evidently, nonlinear classifiers are more capable in fitting training data than linear classifiers. However, nonlinear fusion schemes do not produce significantly better performance on the validation partition of FRGC 3D dataset. One possible reason for this is that there is only neutral expressions in the training partition while expression changes exist in the validation partition.

Table 5.3 and 5.4 summarize the results of two nonlinear fusion schemes, namely polynomial SVM and RBF SVM. At a false accept rate of 0.1%, nonlinear fusions do not yield significantly better performances than those of linear fusions. Note that fusion by RBF SVM results in performance degradation when the number of regions is high, combining 9 or 10 regions in particular.

5.2.6 Discussion

The experiments described above suggest that the LDA method performs better than the sum rule. For ROC III, the Verification rate of LDA and sum rule are 90.04% and 87.96%, respectively (at FAR = 0.1%). The LDA takes the distribution of training data into account so that we can construct a better classifier. By doing fusion at the score level, the FAR can be fixed and then the



(a) Polynomial SVM

(b) RBF SVM

Figure 5.6 Nonlinear fusions of region #1 (*horizontal axis*) and region #5 (*vertical axis*). Decision boundaries (*black lines*) and marginal boundaries (*green and blue lines*) are created by using training partition of FRGC v2.0 dataset. (a) Polynomial SVM, (b) RBF SVM.

Table 5.3 The Verification Rates of Fusion by Polynomial SVM

Number of regions	VR @ FAR = 0.1 %		
	ROC-I	ROC-II	ROC-III
2 regions	.8617	.8540	.8473
3 regions	.8822	.8741	.8652
4 regions	.8846	.8784	.8701
5 regions	.9122	.9056	.8980
6 regions	.9139	.9085	.9025
7 regions	.9141	.9077	.9011
8 regions	.9165	.9105	.9036
9 regions	.9138	.9073	.9005
10 regions	.9131	.9063	.8982

Table 5.4 The Verification Rates of Fusion by RBF SVM

Number of regions	VR @ FAR = 0.1 %		
	ROC-I	ROC-II	ROC-III
2 regions	.8607	.8534	.8453
3 regions	.8810	.8734	.8641
4 regions	.8825	.8754	.8672
5 regions	.9124	.9057	.8985
6 regions	.9117	.9065	.9000
7 regions	.9130	.9068	.8998
8 regions	.9062	.9025	.8978
9 regions	.8772	.8823	.8854
10 regions	.8667	.8713	.8761

verification rate can be computed in both cases. For FRGC experiment 3s, the proposed algorithm achieves similar performance as the best results in the recent FRGC report [121].

Figure 5.7 and 5.8 show the distributions of matching scores for the selected regions before score-level fusion. The score distributions of combining 10 regions by sum rule and LDA are shown in Figure 5.9. It can be seen from this figure that the separability of two classes has been improved as compared to that of the classifiers based on single-regions. The result indicates that the multi-region system performs better than any of the single-region systems.

Intuitively, facial features are different in terms of their importance in face recognition. To explore the relative importance of each region, we began with 10 regions and then leave one region out at each experimental run. After removing one region, we compute the optimal weights on remaining regions by LDA/LSVM. Table 5.1 and 5.2 contains the weights obtained by LDA and LSVM respectively. We first observe that using overlapped regions does not improve performance, *i.e.* region 9 and 10 do not have significant contribution. These tables show a clear difference in the values assigned to each region. The results suggest that nose has the highest contribution followed by forehead. Note that the combination of nose and forehead can achieve a verification rate of 84% at false accept rate of 0.1%. This is a striking new evidence since psychological experiments typically indicate that eyes are the most important followed by mouth and the nose [122].

How could we reasonably explain this finding? One possibility is that the human visual system relies on the projection of a face , *i.e.* 2D texture image, rather than face itself, which is a 3D object, in identifying a face. This new evidence might also explain why and how 3D face recognition can help improve the performance of 2D face recognition systems. The outputs of 2D face recognition and 3D face recognition may not agree with each other when classifying a face because they are dominated by different facial regions. Hence, the error made by one of them might possibly be covered by the other.

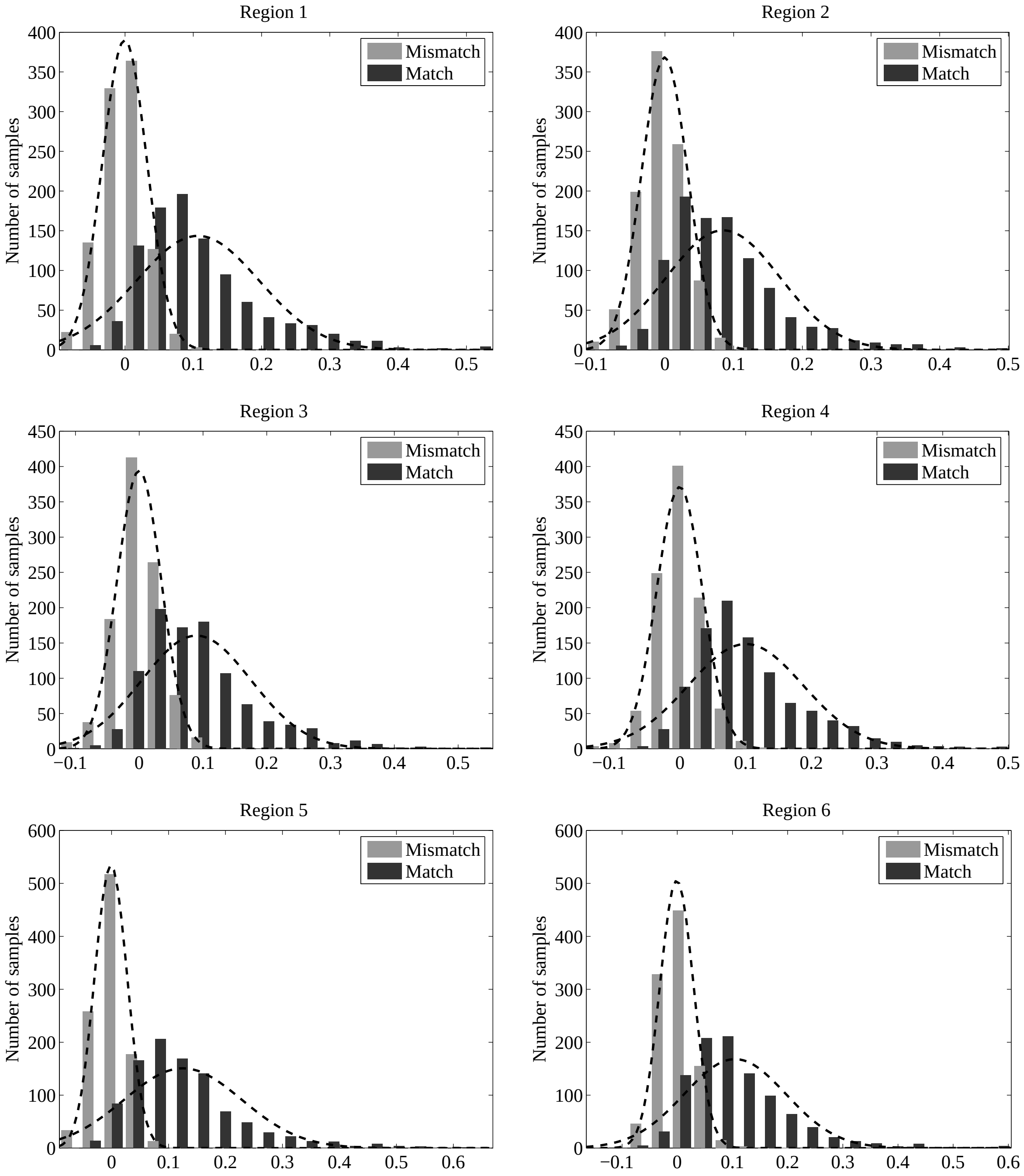


Figure 5.7 Score distribution of region 1,2,3,4,5,6. Higher score means more similar. (dashed lines denote the Gaussian approximation of score distribution).

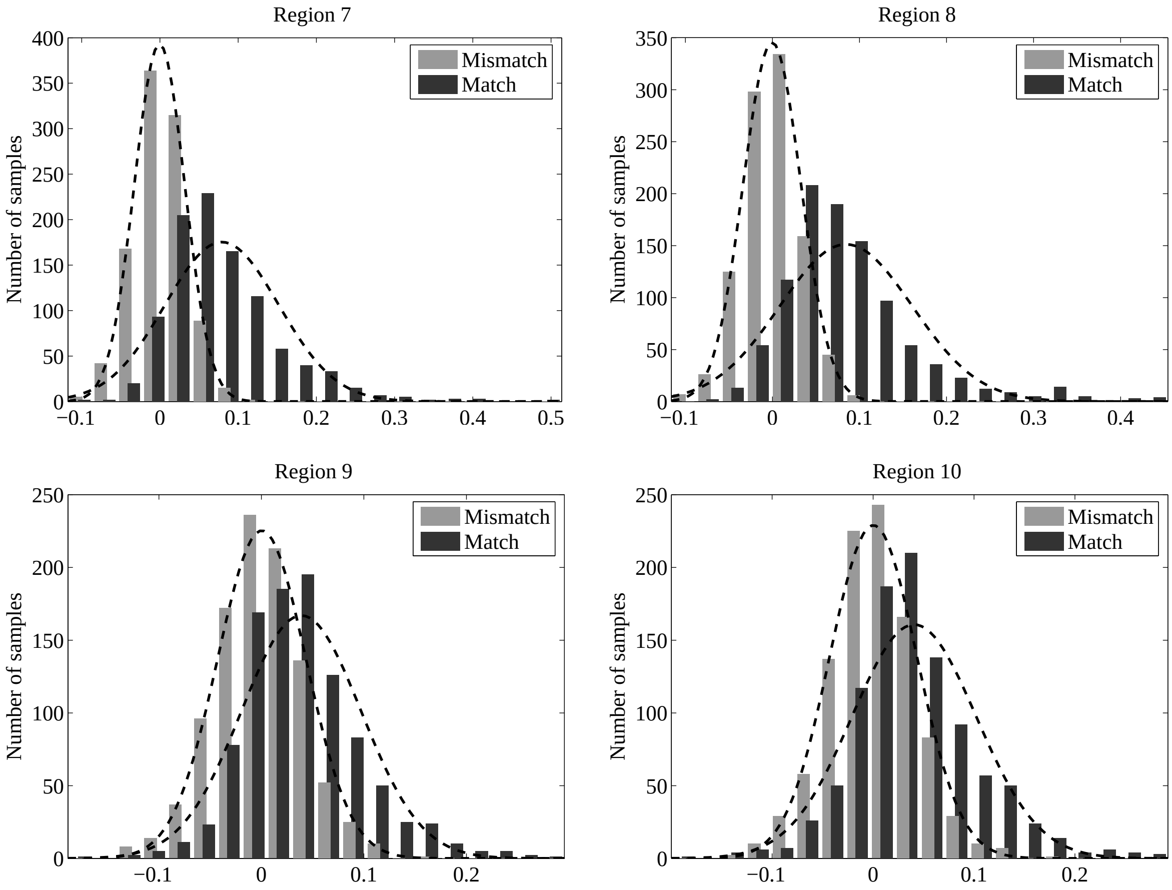
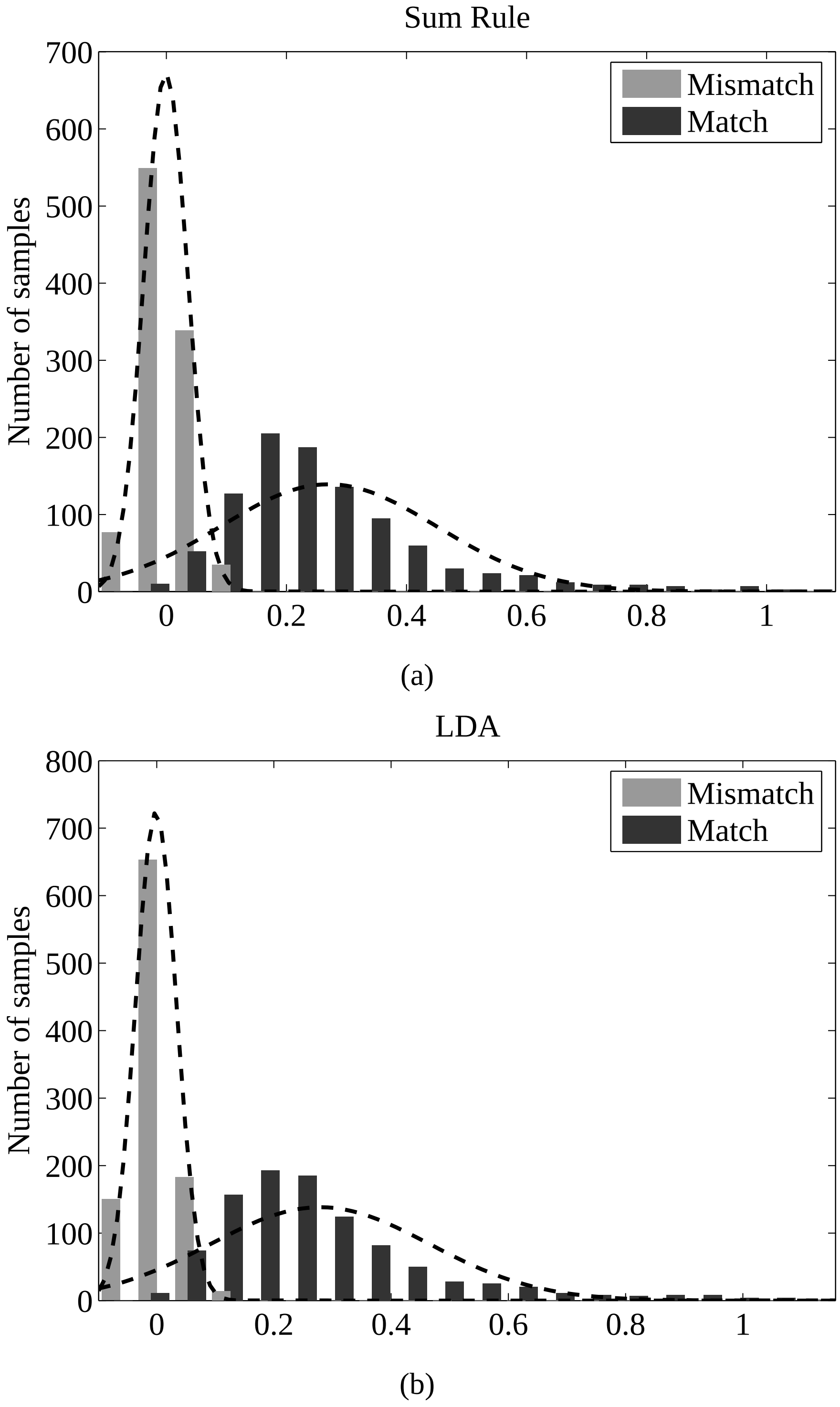


Figure 5.8 Score distribution of region 7,8,9,10. Higher score means more similar. (dashed lines denote the Gaussian approximation of score distribution).

5.2.7 Sensitivity to Facial Expressions

In general, facial expressions could be classified by types (smile, angry, etc.) or strength (neutral, weak, strong). For the purpose of recognition, we are concerned about the performance at different levels of expression strengths, not the types. In this experiment, facial expressions are classified into 3 categories: neutral, small and large. The *neutral* category includes the face postures without obvious expressions, while *small* means natural expressions such as moderate smiles. The *large* category contains the faces with extreme expressions. In many verification/identification scenarios, it is reasonable to assume cooperation from subjects. If a subject is making a weird expression, we can ask him/her to be more cooperative.



(a)

(b)

Figure 5.9 Score distribution of combining 10 regions. (a) Sum rule. (b) LDA.

The validation partition of the FRGC v2.0 dataset, containing 4007 images, is manually classified into these three categories. The manual labeling of images were performed by Geometrix Inc. [123] and the labeling results are available to the research community. The percentage of neutral, small and large expressions are 60%, 20% and 20%, respectively. Figure 5.10 shows the ROC curves of the following experimental settings (from top to bottom): First, neutral images are matched with neutral images (60% vs 60%). Second, neutral images are matched with the union of neutral and small categories (60% vs 80%). Third, neutral images are matched with all 4007 images (60% vs 100%). Fourth, all 4007 images are matched with all 4007 images (100% vs 100%).

From these experiments, we observe that performance degradation is related to the strength of expressions. With neutral and small expressions, our algorithm can achieve verification rate of 96.90% at 0.1% false accept rate (the second curve from top). This is the performance that can be expected with cooperative subjects, *i.e.* natural expressions are allowable but not exaggerated ones. The lower two curves show significant performance degradation due to extreme expressions.

5.3 Summary

A novel 3D face recognition system, which integrates multiple regions of a facial surface, is proposed. The proposed system overcomes the limitations of single-region algorithms. The performance improvement is due to the combination schemes which generate a final matching score with a higher quality than those based on a single region. Experiments indicate that the LDA method performs better than the sum rule. The reason is that the single-region sub-systems do not have similar performance. If the individual classifiers do not have equal performance, we should assign weights to them. The LDA method provides optimal weights for combining multiple classifiers resulting in better performance than the sum rule. LSVM provides another optimal solution for information fusion. The fusion by LSVM achieves similar recognition performance

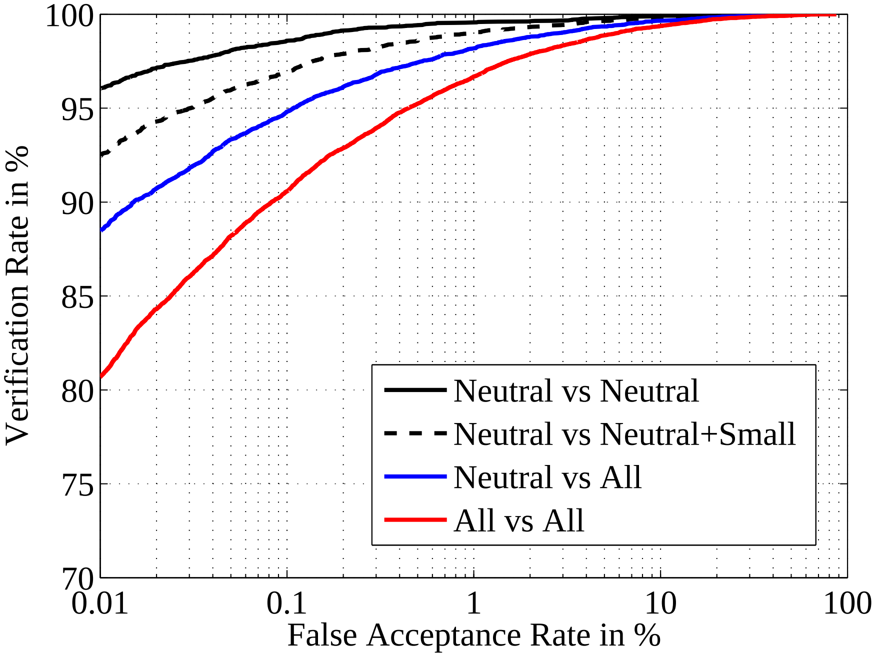


Figure 5.10 ROC performance for different levels of expression strength.

as by LDA, but the LSVM is computationally more intensive. Results on the FRGC v2.0 dataset demonstrate that our multiple-region algorithm is robust in the presence of facial expressions.

Chapter 6

Conclusion and Future Works

6.1 Summary of the Work

This dissertation focused on the issue of feature extraction and presents a novel family of geometrically invariant features. In general, a pattern recognition problem divides into two parts : feature extraction and classification. The goal of feature extraction is to find a good representation so that classification can proceed easily. Unlike classification, which is usually problem-independent, feature extraction is closely related to the problem of interest. For example, a good feature in efforts to identify fingerprints would be very likely of little use in efforts to recognize hand-written letters. In order to develop an efficient and reliable feature extraction scheme, knowledge of the problem domain is required. Hence, we discussed many important cases, including 2D and 3D shape recognition, and demonstrate performance improvement by using the proposed features.

The following four points summarize the key contributions in my dissertation:

- In chapter 2, we presented a systematic approach for constructing invariant features. We detailed four important cases: namely, curve under rigid transformation, curve under affine transformation, surface under rigid transformation, and surface under affine transformation. For each case, we explicitly derived the corresponding summation invariants. We also presented an analytical derivation of the mean squared error for a simple case. The error analysis served mainly to shed light on what one can expect from the calculation of summation invariants in the presence of noise. On the basis of error analysis and the numerical simulation, we observed that the summation invariants can achieve acceptable precision in practical situations.

- In chapter 3, we addressed the problem of 2D shape matching. In the literature [33–36], researchers have commonly used curvature as a geometric feature. However, theoretical analysis [25, 26] also has shown that accurate estimation of curvature can occur only under favorable conditions. Therefore, it is highly desirable to have a reliable feature that can replace the role of curvature in general 2D shape analysis problems.

  We first demonstrated that summation invariants have a better discriminating ability and noise immunity than other shape representations. These other shape representations are wavelet invariants and integral invariants. We used shape retrieval as an example to show that, without other modification, summation invariants can replace curvature.

- Researchers have invested a substantial amount of effort in studying the problem of automatic face recognition. In chapter 4, we approached the challenges in face recognition by introducing summation invariants. We first developed a face recognition algorithm that combines the FRGC baseline algorithm with summation invariant features. In the proposed algorithm, representation of a facial surface uses either a curve Euclidean summation invariant, $\eta_{i,j}$, or a surface Euclidean summation invariant, $\kappa_{i,j,k}$, which are calculated in a semi-local manner. The advantage of using multiple summation invariant features is the complementary nature of the information being extracted. By extracting features through multiple summation invariants, we significantly improved the integrity of the face recognition algorithm.

- It is reported in the literature [42, 43] that the performance of face recognition algorithms drops dramatically owing to expression changes. We approached the issue of facial expression variations in 3D face recognition by using smaller facial regions that are relatively rigid across different expressions. Also, adaptive fusion of multiple regions offers an efficient tool for improvements to classification performance. For example, the mouth region suffers severe performance degradation owing to smiling, but the forehead region is unaffected. In chapter 5, we presented a multi-region face recognition algorithm and showed how it can improve robustness in the presence of facial expressions. We studied several fusion techniques and compared their performances. We validated the proposed algorithm by using the FRGC 3D dataset, totaling 4950 3D scans of 466 subjects. The proposed multi-region face recognition algorithm performed significantly better than the PCA-based FRGC baseline algorithm in situations where the probe set contained neutral and non-neutral expressions.

6.2 Future works

We conclude this dissertation by suggesting some directions along which the research presented here might be extended.

- **2D shape alignment**: In many of the current 2D shape alignment algorithms, such as [1,40], there are two serious drawbacks. First, curvature is used as a feature in the problem formulation , and this state-of-affairs is self-contradictory. More specifically, the goal of 2D shape alignment is to establish nonrigid correspondence, but the curvature is invariant only under rigid transformation. It would be more suitable to use affine invariants, which allow for linear deformation in the case of nonrigid alignment. In this dissertation, we introduced a set of affine summation invariants that would naturally fit in the algorithms for nonrigid alignment. Second, finding the optimal alignment is based on L_2 norm optimization, which resulted in local extrema. So, one has to find the optimal solution by conducting an exhaustive search, using a technique like dynamic programming. It has been shown in [124] that many vision problems can be modified as L_∞ optimization problems and the corresponding cost functionals have a single extremum. By formulating curve alignment as a L_∞ optimization problem, one could find the optimal solution more efficiently without having to conduct an exhaustive search.

- **Fully automatic 3D face recognition**: Currently, our face recognition algorithm relies on manually selected fiducial points. It is highly desirable that, without any human interference, a face recognition algorithm can automatically recognize faces. There are many approaches to automatic face alignment and feature extraction. However, the effects that would arise from an addition of automatic alignment to the proposed algorithm remain a matter for further investigation.

- **Facial Expressions**: The analysis and the modeling of facial expressions have attracted lots of attention [125–127]. These works focus mainly on expression recognition: *i.e.* identifying facial expressions such as those that convey happiness, anger, or disgust. On the other hand, face recognition under expression variations is still an open problem and relatively unaddressed so far. Even though expression recognition and face recognition under expression variations are two different topics, their results could be complementary to each other. For example, knowledge of expression modeling could yield useful information for the development of an expression-invariant face recognition algorithm. Furthermore, the way in which a person poses an expression presents another source of face biometrics. How to use temporal information embedded in facial expressions is an interesting topic for further investigation.

- **Face recognition under partial occlusion**: 3D face recognition is promising because it can help researchers to avoid many of the 2D image-based recognition system's challenging issues, such as pose and illumination variations. However, most of the current 3D face recognition systems require the 3D scan of a facial surface to be a frontal view, which is not feasible in many real world applications. To make the proposed face recognition system more practical, we need to revise many building blocks in our current system, especially those that rely on the frontal-view assumption. After this revision, the input would be a 3D facial scan taken at an arbitrary viewpoint, and the system could match a partial 3D scan against the whole gallery set. Of course, to develop this kind of system, we need a dataset of 3D facial surfaces with large pose changes. Unfortunately, the publicly available 3D datasets contain only frontal view 3D scans. As an important extension of the current project, we would like to submit a proposal for purchasing our own 3D scanner.

LIST OF REFERENCES

[1] E. G. M. Petrakis, A. Diplaros, and E. Milios, “Matching and retrieval of distorted and occluded shapes using dynamic programming,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 11, pp. 1501–1516, 2002.

[2] D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice Hall Professional Technical Reference, 1982.

[3] E. Cartan, “La methode du repere mobile, la theorie des groupes continus, et les espaces generalises,” *Exposes de Geometrie*, no. 5, 1935.

[4] E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker, “Differential and numerically invariant signature curves applied to object recognition,” *Intl. Journal of Computer Vision*, vol. 26, no. 2, pp. 107–135, 1998.

[5] O. Faugeras, “Cartan’s moving frame method and its application to the geometry and evolution of curves in the euclidean, affine and projective planes,” *Lecture Notes in Computer Science*, no. 825, pp. 11 – 46, 1994.

[6] D. Forsyth, J. L. Mundy, A. Zisserman, and C. M. Brown, “Invariance–a new framework for vision,” in *Proc. 3rd Intl. Conf. on Computer Vision*, Osaka, Japan, 1990, pp. 598–605.

[7] M. K. Hu, “Visual pattern recognition by moment invariants,” *IRE – Trans. on Information Theory*, vol. IT-8, no. 2, pp. 179–187, 1962.

[8] T. Reiss, “The revised fundamental theorem of moment invariants,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 830 – 4, 1991.

[9] K. Arbter, W. Snyder, H. Burkhardt, and G. Hirzinger, “Application of affine-invariant fourier descriptors to recognition of 3-d objects,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 640 – 7, 1990.

[10] M. Khalil and M. Bayoumi, “A dyadic wavelet affine invariant function for 2d shape recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1152 – 1164, 2001.

[11] C. E. Hann and M. S. Hickman, “Projective curvature and integral invariants,” *Acta Applicandae Mathematicae*, vol. 74, no. 2, pp. 177–193, 2002.

[12] T. Moons, E. J. Pauwels, L. J. V. Gool, and A. Oosterlinck, “Foundations of semi-differential invariants,” *Intl. Journal of Computer Vision*, vol. 14, no. 1, pp. 25–47, 1995.

[13] E. J. Pauwels, T. Moons, L. J. V. Gool, P. Kempenaers, and A. Oosterlinck, “Recognition of planar shapes under affine distortion,” *Intl. Journal of Computer Vision*, vol. 14, no. 1, pp. 49–65, 1995.

[14] I. Weiss, “Noise-resistant invariants of curves,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 943 – 948, 1993. [Online]. Available: http://dx.doi.org/10.1109/34.232081

[15] A. M. Bruckstein and A. N. Netravali, “On differential invariants of planar curves and recognizing partially occluded planar shapes,” *Annals of Mathematics and Artificial Intelligence*, vol. 13, no. 3-4, pp. 227–50, / 1995.

[16] J. Flusser and T. Suk, “Pattern recognition by affine moment invariants,” *Pattern Recognition*, vol. 26, no. 1, pp. 167 – 74, 1993/01/.

[17] S. X. Liao and M. Pawlak, “On image analysis by moments,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 254 – 266, 1996.

[18] J. Sato and R. Cipolla, “Affine integral invariants and matching of curves,” in *Proc. of 13th Intl. Conf. on Pattern Recognition*, vol. 1, 1996, pp. 915–19.

[19] M. Fels and P. J. Olver, “Moving coframes: I. a practical algorithm,” *Acta Applicandae Mathematicae*, vol. 51, no. 2, pp. 161 – 213, 1998.

[20] ——, “Moving coframes: Ii. regularization and theoretical foundations,” *Acta Applicandae Mathematicae*, vol. 55, no. 2, pp. 127 – 208, 1999.

[21] L. V. Gool, T. Moons, E. Pauwels, and A. Oosterlinck, *Geometric invariance in computer vision*. Cambridge: MIT, 1992, ch. Semi-Differential Invariants, pp. 157–192.

[22] I. Weiss, “Geometric invariants and object recognition,” *International Journal of Computer Vision*, vol. 10, no. 3, pp. 207–31, 06 1993.

[23] S. Manay, B.-W. Hong, A. Yezzi, and S. Soatto, “Integral invariant signatures,” *ECCV 2004. Proc. (Lecture Notes in Comput. Sci. Vol.3024)*, vol. 4, pp. 87 – 99, 2004.

[24] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[25] S. Utcke, “Error-bounds on curvature estimation,” in *Scale Space*, British Machine Vision Association. Isle of Skye, Scotland, UK: Springer-Verlag, Berlin, June 2003, pp. 657–666. [Online]. Available: http://kogs-www.informatik.uni-hamburg.de/ utcke/Papers/utcke:03:ss.pdf

[26] V. Kovalevsky, “Curvature in digital 2d images.” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 7, pp. 1183–1200, 2001.

[27] N. Ayache and O. D. Faugeras, “Hyper: a new approach for the recognition and positioning to two-dimensional objects,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 44–54, 1986.

[28] E. Milios and E. Petrakis, “Shape retrieval based on dynamic programming,” *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 141–146, 2000.

[29] Y. Gdalyahu and D. Weinshall, “Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1312–1328, 1999.

[30] S. D. Connell and A. K. Jain, “Learning prototypes for on-line handwritten digits,” in *ICPR ’98: Proceedings of the 14th International Conference on Pattern Recognition-Volume 1*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 182–4.

[31] B. Wirtz, “Average prototypes for stroke-based signature verification,” in *ICDAR ’97: Proceedings of the 4th International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 268–272.

[32] S. Umeyama, “Parameterized point pattern matching and its application to recognition of object families,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 2, pp. 136–144, 1993.

[33] I. Cohen, N. Ayache, and P. Sulger, “Tracking points on deformable objects using curvature information,” in *ECCV ’92: Proceedings of the Second European Conference on Computer Vision*. London, UK: Springer-Verlag, 1992, pp. 458–466.

[34] H. Tagare, “Shape-based nonrigid correspondence with application to heart motion analysis.” *IEEE Trans. Med. Imaging*, vol. 18, no. 7, pp. 570–579, 1999.

[35] R. Basri, L. Costa, D. Geiger, and D. Jacobs, “Determining the similarity of deformable shapes,” Weizmann Science Press of Israel, Jerusalem, Israel, Israel, Tech. Rep. CS98-05, 1998.

[36] L. Younes, “Computable elastic distances between shapes,” *SIAM Journal on Applied Mathematics*, vol. 58, no. 2, pp. 565–586, 1998.

[37] F. Mokhtarian, S. Abbasi, and J. Kittler, “Indexing an image database by shape content using curvature scale space,” *IEE Colloquium (Digest)*, no. 119, pp. 4 – 1, 1996.

[38] T. Pavlidis, “A review of algorithms for shape analysis,” pp. 145–160, 1995.

[39] S. Loncaric, “A survey of shape analysis techniques,” *Pattern Recognition*, vol. 31, no. 8, pp. 983–1001, 1998. [Online]. Available: citeseer.ist.psu.edu/loncaric98survey.html

[40] T. B. Sebastian, P. N. Klein, and B. B. Kimia, “On aligning curves,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 116–125, 2003.

[41] N. Ueda and S. Suzuki, “Learning visual models from shape contours using multiscale convex/concave structure matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 337–352, 1993.

[42] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The feret evaluation methodology for face-recognition algorithms,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–104, 2000.

[43] P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, “Face recognition vendor test 2002,” in *Intl. Workshop on Analysis and Modeling of Faces and Gestures*, 2003, p. 44.

[44] K. I. Chang, K. W. Bowyer, and P. J. Flynn, “An evaluation of multimodal 2d+3d face biometrics,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 619–24, 04 2005.

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

[46] D. J. Field, “What is the goal of sensory coding?” *Neural Comput.*, vol. 6, no. 4, pp. 559–601, 1994.

[47] J. Daugman, “Two-dimensional spectral analysis of cortical receptive field profiles,” *Vision Research*, vol. 20, pp. 847–856, 1980.

[48] ——, “Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters,” *J. Opt. Soc. Am.*, vol. 2, no. 7, pp. 1160–1169, 1985.

[49] J. Jones and L. Palmer, “An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex,” *J. Neurophysiology*, pp. 1233–1258, 1987.

[50] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen, “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Transactions on Computers*, vol. 42, pp. 300–311, 1993.

[51] C. Liu and H. Wechsler, “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition.” *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 467–476, 2002.

[52] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Classifying facial actions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999.

[53] C. Liu and H. Wechsler, "Robust coding schemes for indexing and retrieval from large face databases." *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 132–137, 2000.

[54] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.

[55] B. Moghaddam, C. Nastar, and A. Pentland, "A bayesian similarity measure for direct image matching," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 2, 1996, pp. 350–358.

[56] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[57] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images (invited paper)," in *AVBPA '97: Proceedings of the First International Conference on Audio- and Video-Based Biometric Person Authentication*. London, UK: Springer-Verlag, 1997, pp. 127–142.

[58] C. Liu and H. Wechsler, "A shape and texture based enhanced fisher classifier for face recognition," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 598–608, 2001.

[59] M. Bartlett, J. Movellan, and T. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on neural networks*, vol. 13, no. 6, pp. 1450–1464, 2002.

[60] C. Liu and H. Wechsler, "Comparative assessment of independent component analysis (ICA) for face recognition," in *International Conference on Audio and Video-based Biometric Person Authentication*, 1999, pp. 211–216.

[61] K. W. Bowyer, 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.

[62] J. Y. Cartoux, J. T. Lapreste, and M. Richetin, "Face authentification or recognition by profile extraction from range images," in *Proc. of Workshop on Interpretation of 3D Scenes*, 1989, pp. 194–9.

[63] J. C. Lee and E. Milios, "Matching range images of human faces," in *Proc. of Third Intl. Conf. on Computer Vision*, 1990, pp. 722–6.

[64] G. G. Gordon, "Face recognition based on depth and curvature features," in *Proc. IEEE Conf. on CVPR*, 1992, pp. 808–10.

[65] H. T. Tanaka, M. Ikeda, and H. Chiaki, "Curvature-based face surface recognition using spherical correlation. principal directions for curved object recognition," in *Proc. of Third IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, 1998, pp. 372–7.

[66] Y. hak Lee; Jae-chang Shim, “Curvature based human face recognition using depth weighted hausdorff distance,” in *International Conference on Image Processing (ICIP ’04)*, vol. 3, 2004, pp. 1429– 1432.

[67] Y. Lee, H. Song, U. Yang, H. Shin, and K. Sohn, “Local feature based 3d face recognition,” in *LNCS 3546: 5th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA 2005)*, 2005, pp. 909–918.

[68] A. B. Moreno, Ángel Sánchez, J. F. Vélez, and F. J. Díaz, “Face recognition using 3d surface-extracted descriptors,” in *Proceedings of the Irish Machine Vision and Image Processing (IMVIP ’03)*, 2003.

[69] G. Medioni and R. Waupotitsch, “Face modeling and recognition in 3-d,” in *2003 IEEE Intl. Workshop on Analysis and Modeling of Faces and Gestures*, 2003, pp. 232–3.

[70] C. Hesher, A. Srivastava, and G. Erlebacher, “A novel technique for face recognition using range imaging,” in *Proc. of the Seventh Intl. Symposium on Signal Processing and its Applications*, vol. 2, 2003, pp. 201–4.

[71] K. I. Chang, K. W. Bowyer, and P. J. Flynn, “Multimodal 2d and 3d biometrics for face recognition,” in *2003 IEEE Intl. Workshop on Analysis and Modeling of Faces and Gestures*, 2003, pp. 187–94.

[72] T. Heseltine, N. Pears, and J. Austin, “Three-dimensional face recognition: An eigensurface approach,” in *Proc. of the Intl. Conf. on Image Processing*, vol. 5, 2004, pp. 1421–1424.

[73] I. A. Kakadiaris, G. Passalis, T. Theoharis, G. Toderici, I. Konstantinidis, and N. Murtuza, “Multimodal face recognition: Combination of geometry with physiological information,” in *Proc. IEEE Conf. on CVPR*, vol. 2, 2005, pp. 1022–1029.

[74] K. I. Chang, K. Bowyer, and P. J. Flynn, “Adaptive rigid multi-region selection for handling expression variation in 3d face recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops*, 2005, p. 157.

[75] G. Passalis, I. A. Kakadiaris, T. Theoharis, G. Toderici, and N. Murtuza, “Evaluation of 3d face recognition in the presence of facial expressions: an annotated deformable model approach,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops*, 2005, p. 171.

[76] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, “Expression-invariant 3d face recognition,” in *Proc. of the 4th Intl. Conf. on Audio- and Video-Based Biometric Person Authentication*, 2003, pp. 62–9.

[77] ——, “Three-dimensional face recognition,” *Int. J. Comput. Vision*, vol. 64, no. 1, pp. 5–30, 2005.

[78] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3d faces,” in *SIGGRAPH ’99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[79] A. Martinez and R. Benavente, “The ar face database,” Computer Vision Center (CVC) Technical Report, Tech. Rep., 1998.

[80] E. Bailly-Bailliére, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, B. Ruíz, and J.-P. Thiran, “The banca database and evaluation protocol.” in *Proceedings of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, 2003, pp. 625–638.

[81] T. Sim, S. Baker, and M. Bsat, “The cmu pose, illumination, and expression database,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615 – 1618, December 2003.

[82] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, “Overview of the face recognition grand challenge,” in *Proc. IEEE Conf. on CVPR*, vol. 1, 2005, pp. 947–54.

[83] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[84] N. Troje and H. H. Blthoff, “Face recognition under varying poses: The role of texture and shape.” *Vision Research*, vol. 36, pp. 1761–1771, 1996.

[85] National Institute of Standards and Technology, “NIST special database 18: Mugshot identification database (MID),” available at http://www.nist.gov/srd/nistsd18.htm.

[86] K. Messer, J. Matas, J. Kittler, and K. Jonsson, “XM2VTSDB: The extended M2VTS database,” in *Audio- and Video-based Biometric Person Authentication (AVBPA)*, 1999, pp. 72–77.

[87] S.-Y. Kung and M.-W. Mak, “On consistent fusion on multimodal biometrics,” in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’06)*, Tououlse, France, 2006.

[88] G. Pan, S. Han, Z. Wu, and Y. Wang, “3d face recognition using mapped depth images,” in *IEEE Workshop on Face Recognition Grand Challenge Experiments*, June 2005.

[89] T. Russ, M. Koch, and C. Little, “A 2d range hausdorff approach for 3d face recognition,” in *IEEE Workshop on Face Recognition Grand Challenge Experiments*, June 2005.

[90] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On combining classifiers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[91] L. Hong and A. K. Jain, “Integrating faces and fingerprints for personal identification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1295–1307, 1998.

[92] A. Ross and A. Jain, “Information fusion in biometrics,” *Pattern Recogn. Lett.*, vol. 24, no. 13, pp. 2115–2125, 2003.

[93] D. H. Wolpert, “Stacked generalization,” *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.

[94] S. Fine, R. Gilad-Bachrach, and E. Shamir, “Query by committee, linear separation and random walks,” *Theor. Comput. Sci.*, vol. 284, no. 1, pp. 25–51, 2002.

[95] Y. Freund, “Boosting a weak learning algorithm by majority,” *Inf. Comput.*, vol. 121, no. 2, pp. 256–285, 1995.

[96] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Computational Learning Theory*, 1992, pp. 287–294. [Online]. Available: citeseer.ist.psu.edu/seung92query.html

[97] J. Twomey and A. Smith, “Committee networks by resampling,” 1995. [Online]. Available: citeseer.ist.psu.edu/twomey95committee.html

[98] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik, “Boosting and other ensemble methods,” *Neural Computat.*, vol. 6, pp. 1289–1301, 1994.

[99] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, 1990.

[100] A. Krogh and J. Vedelsby, “Neural network ensembles, cross validation, and active learning,” *G. Tesauro, D. Touretzky, and T. Leen, editors, Advances in Neural Information Processing Systems*, vol. 7, pp. 231–238, 1995.

[101] M. Perrone and L. N. Cooper, “When networks disagree: ensemble method for neural networks,” *Artificial Neural Networks for Speech and Vision, R. J. Mammone, Ed.*, pp. 126–142, 1993.

[102] P. Sollich and A. Krogh, “Learning with ensembles : How over-fitting can be useful,” *Advances in Neural Information Processing Systems*, vol. 8, pp. 190–196, 1996.

[103] K. Tumer and J. Ghosh, “Error correlation and error reduction in ensemble classifiers,” *Connection Science*, vol. 8, no. 3-4, pp. 385–403, 1996. [Online]. Available: citeseer.ist.psu.edu/tumer96error.html

[104] M. Jordan and R. Jacobs, “Hierarchical mixtures of experts and the EM algorithm,” *IJCNN ’93-Nagoya. Proceedings of 1993 International Joint Conference on Neural Networks (Cat. No.93CH3353-0)*, vol. 2, pp. 1339 – 44, 1993.

[105] L. Xu, M. I. Jordan, and G. E. Hinton, “An alternative model for mixture of experts,” *Advances in Neural Information Processing Systems*, pp. 633–640, 1995.

[106] M. Jordan and L. Xu, “Convergence properties of the em approach to learning in mixture-of-experts architectures,” *Technical Report 9302, Department of Brain and Cognitive Science, MIT*, 1993.

[107] Z. Chair and P. Varshney, “Optimal data fusion in multiple sensor detection systems,” *Multidimensional Systems and Signal Processing*, vol. AES-22, no. 1, pp. 9–31, 1986.

[108] B. Chen and P. Varshney, “A bayesian sampling approach to decision fusion using hierarchical models,” *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1809–1818, Aug. 2002.

[109] L. Xu, A. Krzyzak, and C. Y. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418–435, May 1992.

[110] K. Woods, W. P. Kegelmeyer Jr., and K. Bowyer, “Combination of multiple classifier using local accuracy estimates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, Apr. 1997.

[111] C. Ji and S. Ma, “Combinations of weak classifiers,” *IEEE Transaction on Neural Networks*, vol. 8, no. 1, pp. 32–42, Jan. 1997.

[112] M. Petrakos, J. A. Benediktsson, and I. Kanellopoulos, “The effect of classifier agreement on the accuracy of the combined classifier in decision level fusion,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 11, pp. 2539–2546, Nov. 2001.

[113] Y. Huang and C. Suen, “A method of combining multiple experts for the recognition of unconstrained handwritten numerals,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, Jan. 1995.

[114] M. Faundez-Zanuy, “Data fusion in biometrics,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 1, pp. 34–38, 2005.

[115] J. J. Clark and A. L. Yuille, *Data Fusion for Sensory Information Processing Systems*. Norwell, MA, USA: Kluwer Academic Publishers, 1990.

[116] L. I. Kuncheva, C. J. Whitaker, C. A. Shipp, and R. P. W. Duin, “Is independence good for combining classifiers?” in *ICPR*, 2000, pp. 168–171.

[117] W.-Y. Lin, K.-C. Wong, N. Boston, and Y. H. Hu, "Fusion of summation invariants in 3d human face recognition," in *to appear in the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006.

[118] D. G. S. Richard O. Duda, Peter E. Hart, *Pattern Classification*, 2nd ed. Wiley Interscience, 2000.

[119] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[120] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[121] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, and W. Worek, "Preliminary face recognition grand challenge results," in *7th International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 15–24.

[122] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: 20 results all computer vision researchers should know about," *(under review)*, available at http://web.mit.edu/bcs/sinha/publications.html.

[123] T. Maurer, D. Guigonis, I. Maslov, B. Pesenti, A. Tsaregorodtsev, D. West, and G. Medioni, "Performance of geometrix activeid 3d face recognition engine on the frgc data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, vol. 3, 2005, pp. 154–154.

[124] R. Hartley and F. Schaffalitzky, "L-infinity minimization in geometric reconstruction problems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR 2004*, vol. I. Washington (DC), USA: IEEE, June 2004, pp. I–504–509.

[125] I. A. Essa and A. P. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 757–763, 1997.

[126] A. Samal and P. A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: a survey," *Pattern Recogn.*, vol. 25, no. 1, pp. 65–77, 1992.

[127] J. Lien, T. Kanade, J. Cohn, and C. Li, "Automated facial expression recognition based on facs action units," in *Third IEEE International Conference on Automatic Face And Gesture Recognition*, 1998.